Prior-preconditioned conjugate gradient for accelerated Gibbs sampling in "large n & large p" sparse Bayesian logistic regression models

Akihiko Nishimura and Marc A. Suchard

University of California - Los Angeles, USA

Summary. In a modern observational study based on healthcare databases, the number of observations typically ranges in the order of $10^5 \sim 10^6$ and that of the predictors in the order of $10^4 \sim 10^5$. Despite the large sample size, the data rarely provide sufficient information to reliably estimate such a large number of parameters. Sparse regression provides a potential solution to this problem. There is a rich literature on desirable theoretical properties of Bayesian approaches based on shrinkage priors. On the other hand, the development of scalable methods for the required posterior computation has largely been limited to the $p \gg n$ case. Shrinkage priors make the posterior amenable to Gibbs sampling, but a major computational bottleneck still arises from the need to sample from a high-dimensional Gaussian distribution at each iteration. Despite a closed-form expression for the precision matrix $\boldsymbol{\Phi}$, computing and factorizing such a large matrix is computationally expensive nonetheless. 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 $\boldsymbol{\Phi\beta} = \boldsymbol{b}$ has the desired Gaussian distribution. We can then find an accurate solution to the linear system by the conjugate gradient (CG) algorithm through the matrix-vector multiplications by $\boldsymbol{\Phi}$, without ever explicitly inverting $\boldsymbol{\Phi}$. Practical performance of CG, however, depends critically on appropriate *preconditioning* of the linear system; we turn CG into a highly effective algorithm for sparse Bayesian regression by developing a theory of *prior-preconditioning*. We apply our algorithm to a clinically relevant large-scale observational study with $n = 72{,}489$ and $p = 22{,}175$, designed to assess the relative risk of intracranial hemorrhage from two alternative blood anti-coagulants. Our algorithm demonstrates an order of magnitude speed-up in the posterior computation.

Keywords: Bayesian, Markov chain Monte Carlo, conjugate gradient, numerical linear algebra, sparsity, high-dimensional inference

1. Introduction

Given an outcome of interest $\boldsymbol{y} \in \{0,1\}^n$ and a large number of features $\boldsymbol{x}_1, \dots \boldsymbol{x}_p$, the goal of sparse regression (or variable selection) 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 when either

(a) the design matrix $\boldsymbol{X}$ is sparse i.e. only a small fraction of the design matrix contains non-zero entries because the features are observed infrequently.

(b) the outcome $\boldsymbol{y}$ is imbalanced i.e. $y_i = 0$ (or $y_i = 1$) for most of $i = 1, \ldots, n$.

Either of these conditions reduces the amount of information the data provides on the regression coefficients. In a modern observational study based on large-scale healthcare databases, for example, the sparsity of $\boldsymbol{X}$ almost always holds true because a large number of potential pre-existing conditions and available treatments exist, yet only a small subset of these applies to each patient (Schuemie et al., 2018b). Imbalanced outcome is also common as many serious diseases of interest are rare among the population.

The particular application considered in this manuscript is a comparative study of two anti-coagulants *dabigatran* and *warfarin*, based on observational data from Truven Health MarketScan Medicare Supplemental and Coordination of Benefits Database. These drugs reduce the risk of blood clot formation, but can increase the risk of bleeding. The goal of the study is to quantify which of the two drugs has a lower risk of intracranial hemorrhage (bleeding inside the skull) among treated patients. The data set consists of $n = 72{,}489$ patients and $p = 22{,}175$ predictors of potential relevance.

An increasingly common approach to sparse regression is the Bayesian method based on continuous shrinkage priors on the regression coefficients $\boldsymbol{\beta}$. These priors often are represented as a scale-mixture of Gaussians

$$\beta_j \,|\, \lambda_j, \tau \sim \mathcal{N}(0, \tau^2 \lambda_j^2), \;\; \lambda_j \sim \pi_L(\cdot), \;\; \tau \sim \pi_T(\cdot), \tag{1.1}$$

with unknown *global scale* parameter τ and *local scale* parameter λ_j (Polson and Scott, 2010; Carvalho et al., 2010; Armagan et al., 2013; Polson et al., 2014; Bhattacharya et al., 2015; Bhadra et al., 2017). Compared to more traditional "spike-and-slab" discrete-mixture priors for sparse Bayesian regression, these continuous shrinkage priors are typically more computationally efficient while maintaining highly desirable statistical properties (Bhattacharya et al., 2015; Pal et al., 2014; Datta et al., 2013). Despite the relative computational advantage, however, posterior inference under these priors still encounters a major scalability issue. For instance, for our comparative study of two anti-coagulants, it takes over 50 hours on 2015 iMac to run 1,000 iterations of the current state-of-the-art Gibbs sampler in our reasonably optimized Python implementation (Section 5).

In this article, we focus on accelerating the state-of-the-art Gibbs sampler for sparse Bayesian logistic regression, but our approach applies whenever the likelihood of the data or latent parameter can be expressed as a Gaussian mixture. The Polya-Gamma data augmentation scheme of Polson et al. (2013) makes a posterior under the logistic model amenable to Gibbs sampling as follows. Through a Polya-Gamma auxiliary parameter $\boldsymbol{\omega}$, the conditional likelihood of a binary outcome $\boldsymbol{y}$ becomes

$$y_i' \,|\, \boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\omega} \sim \mathcal{N}(\boldsymbol{x}_i^\intercal \boldsymbol{\beta}, \omega_i^{-1}) \;\; \text{for} \;\; y_i' := \omega_i^{-1} (y_i - 1/2) \,. \tag{1.2}$$

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

$$\boldsymbol{\beta} \,|\, \boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\Phi}^{-1} \boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{y}', \boldsymbol{\Phi}^{-1}) \;\; \text{for} \;\; \boldsymbol{\Phi} = \boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X} + \tau^{-2} \boldsymbol{\Lambda}^{-2}, \tag{1.3}$$

where $\boldsymbol{\Omega} = \text{diag}(\boldsymbol{\omega})$, a diagonal matrix with entries $\Omega_{ii} = \omega_i$, and $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$. The main computational bottleneck of the Gibbs sampler is the need to sample from a high-dimensional Gaussian of the form (1.3). The standard algorithm requires $O(n^2p + p^3)$ operations: $O(n^2p)$ for computing the term $\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}$ for $\boldsymbol{\Phi}$ and $O(p^3)$ for the Cholesky decomposition of $\boldsymbol{\Phi}$. These operations remain significant computational burdens even when a sparsity in $\boldsymbol{X}$ significantly reduces the theoretically necessary number of arithmetic operations. This is because numerical computations in a modern computer architecture is memory bound, and the reduction in the number of arithmetic operations does not directly translate into reduced computing time (Golub and Van Loan, 2012; Dongarra et al., 2016).

Recently, significant progress has been made in computational techniques for $n \ll p$ cases. Bhattacharya et al. (2016) proposes an algorithm to sample from (1.3) with only $O(n^2p + n^3)$ operations, where the $O(n^3)$ cost becomes negligible for $n \ll p$. The algorithm requires computing the $n \times n$ matrix $\boldsymbol{X}\boldsymbol{\Lambda}^2\boldsymbol{X}^\intercal$ for $O(n^2p)$ operations, and then solving an $n \times n$ linear system for $O(n^3)$ operations. Johndrow et al. (2018) reduce the $O(n^2p)$ cost by replacing the matrix $\boldsymbol{X}\boldsymbol{\Lambda}^2\boldsymbol{X}^\intercal$ with an approximation that can be computed with $O(n^2k)$ operations for $k < p$. This technique offers no reduction in computational cost in $n > p$ cases, however. Hahn et al. (2018) propose a sampling approach for linear regression based on an extensive pre-processing of the matrix $\boldsymbol{X}^\intercal\boldsymbol{X}$ — a trick limited in scope strictly to Gaussian likelihood models. None of these advances addresses the "large n & large p" logistic regression problem considered in this article.

Proposed in this article is a novel algorithm to sample from a high-dimensional distribution of the form (1.3) through the conjugate gradient (CG) method, requiring only a small number of the matrix-vector multiplication operations $\boldsymbol{v} \to \boldsymbol{\Phi}\boldsymbol{v}$. The vector $\boldsymbol{\Phi}\boldsymbol{v}$ can be computed through operations $\boldsymbol{v} \to \boldsymbol{X}\boldsymbol{v}$ and $\boldsymbol{w} \to \boldsymbol{X}^\intercal\boldsymbol{w}$ along with simple element-wise multiplications, without ever explicitly forming the matrix $\boldsymbol{\Phi}$. This is an important feature when dealing with a large and sparse design matrix $\boldsymbol{X}$; the matrix $\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}$ and hence $\boldsymbol{\Phi}$ may contain a much larger proportion of non-zero entries than $\boldsymbol{X}$ does, making directly handling $\boldsymbol{\Phi}$ much more intensive in memory usage and the number of arithmetic operations. Note, for example, that it requires 74.5 GB of memory to allocate a $p \times p$ dense matrix in double-precision numbers when $p = 10^5$. The ability to automatically exploits a sparsity structure in $\boldsymbol{X}$, therefore, is another major advantage of our algorithm.

Also developed in this article is a theory of an effective *preconditioning* technique in the context of sparse Bayesian regression. Preconditioning relates a given problem to a modified one to accelerate CG and is critical in achieving the full potential of the algorithm. While a variety of general-purpose preconditioners exist, design of an effective preconditioner remains problem specific. Exploiting the fact that the shrinkage priors dominate likelihood for all but a small fraction of the regression coefficients, we develop what we term the *prior-preconditioning* approach and demonstrate its superiority over general-purpose preconditioners in sparse Bayesian regression applications. It is worth noting that our contribution here is completely distinct from that of Cockayne et al. (2018), who propose an extension of CG that outputs a probability measure. We instead employ the classical CG method in

a novel context, demonstrating its ability to significantly accelerate Monte Carlo simulations when applied in a right way.

The rest of the paper is organized as follows. Section 2 begins by describing how to recast the problem of sampling from the distribution (1.3) as that of solving a linear system $\boldsymbol{\Phi}\boldsymbol{\beta} = \boldsymbol{b}$, eliminating the need to factorize $\boldsymbol{\Phi}$. The rest of the section explains how to apply CG to rapidly solve the linear system while developing theoretical foundations behind our prior-preconditioner. In Section 3, using simulated data, we demonstrate the effectiveness of the CG-based sampler for sparse regressions. Also demonstrated is how the behavior of CG depends on different preconditioning strategies. Section 4 describes practical details needed to successfully apply the CG-based sampler to sparse regression problems. Finally, in Section 5, we apply our algorithm to carry out the dabigatran vs. warfarin comparison, demonstrating an order of magnitude speed-up in the posterior computation. Our CG-accelerated Gibbs sampler for sparse Bayesian logistic regression 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/aki-nishimura/bayes-bridge`.

2. Conjugate gradient sampler for sparse regression

2.1. Generating a Gaussian as the solution of a linear system

The standard algorithm for sampling a multivariate-Gaussian requires the Cholesky factorization $\boldsymbol{\Phi} = \boldsymbol{L}\boldsymbol{L}^\intercal$ of its precision (or covariance) matrix (Ripley, 1987). When the precision matrix $\boldsymbol{\Phi}$ has a specific structure as in (1.3), however, it turns out that the problem of sampling from the distribution (1.3) can be recast to that of solving a linear system. This in particular obviates the need to factorize the matrix $\boldsymbol{\Phi}$. The key observation is that we can generate a Gaussian vector $\boldsymbol{b}$ with $\mathrm{Var}(\boldsymbol{b}) = \boldsymbol{\Phi}$ for a computational cost negligible compared to an explicit formation of $\boldsymbol{\Phi}$.

PROPOSITION 2.1. *The following procedure generates a sample $\boldsymbol{\beta}$ from distribution* (1.3)*:*

(a) Generate $\boldsymbol{b} \sim \mathcal{N}\big(\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{y}', \boldsymbol{\Phi}\big)$ by sampling independent Gaussian vectors $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$ and $\boldsymbol{\delta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ and then setting

$$\boldsymbol{b} = \boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{y}' + \boldsymbol{X}^\intercal\boldsymbol{\Omega}^{1/2}\boldsymbol{\eta} + \tau^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\delta}. \tag{2.4}$$

(b) Solve the following linear system for $\boldsymbol{\beta}$:

$$\boldsymbol{\Phi}\boldsymbol{\beta} = \boldsymbol{b} \quad \big(\textit{ where } \boldsymbol{\Phi} = \boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X} + \tau^{-2}\boldsymbol{\Lambda}^{-2}\big). \tag{2.5}$$

The result follows immediately from basic properties of multivariate Gaussians; in particular, the solution to (2.5) has the required covariance structure because $\mathrm{Var}(\boldsymbol{\Phi}^{-1}\boldsymbol{b}) = \boldsymbol{\Phi}^{-1}\mathrm{Var}(\boldsymbol{b})(\boldsymbol{\Phi}^{-1})^\intercal$.

The above algorithm complements the algorithm Bhattacharya et al. (2016) propose; their algorithm reduces the task of sampling a multivariate Gaussian to solving a $n \times n$ linear system, while our algorithm reduces it to solving a $p \times p$ system. While

the structures of the two linear systems are different, the techniques developed in this manuscript may be applicable with appropriate modifications to solving the $n \times n$ linear system arising in Bhattacharya et al. (2016).

2.2. Iterative method for solving a linear system

The utility of Proposition 2.1 stems from the fact that solving the linear system (2.5) 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, 1996). Significant research has gone into how best to apply iterative methods in this specific context; see Stein et al. (2012), Sun and Stein (2016), and Stroud et al. (2017) for example. Outside the Gaussian process literature, Zhou and Guan (2017) use an iterative method to address the bottleneck of having to solve large linear systems when computing Bayes factors in a model selection problem.

The use of CG as a computational tool for Monte Carlo simulation is a novel feature of our work that has not been considered by any the above works. While we were preparing our manuscript, we were also informed of the work by Zhang et al. (2018), which uses the same idea as in Proposition 2.1. They apply the CG method, apparently without any preconditioners, to generate a posterior sample from a Gaussian process model. Our work is distinguished by a careful development — supported by both theoretical analysis and systematic empirical evaluations — of preconditioning techniques (Section 2.3) tailored toward sparse Bayesian regression problems. The theoretical foundations laid out here also provide practical guidelines on how one may apply CG in other Bayesian computation 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 $\boldsymbol{\beta}_0 = \mathbf{0}$ 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*

$$\mathcal{K}(\boldsymbol{\Phi}, \boldsymbol{r}_0, k) = \text{span}\left\{\boldsymbol{r}_0, \boldsymbol{\Phi}\boldsymbol{r}_0, \ldots, \boldsymbol{\Phi}^{k-1}\boldsymbol{r}_0\right\}, \tag{2.6}$$

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

$$\boldsymbol{\beta}_k = \text{argmin}\left\{\left\|\boldsymbol{\beta}' - \boldsymbol{\beta}\right\|_{\boldsymbol{\Phi}} : \boldsymbol{\beta}' \in \boldsymbol{\beta}_0 + \mathcal{K}(\boldsymbol{\Phi}, \boldsymbol{r}_0, k)\right\} \text{ where } \|\boldsymbol{r}\|_{\boldsymbol{\Phi}}^2 := \boldsymbol{r}^\mathsf{T}\boldsymbol{\Phi}\boldsymbol{r}. \tag{2.7}$$

The norm $\|\cdot\|_{\boldsymbol{\Phi}}$ is often referred to as the $\boldsymbol{\Phi}$-norm. The optimality property (2.7) in particular implies that CG yields the exact solution after p iterations. The

main computational cost of each update $\boldsymbol{\beta}_k \to \boldsymbol{\beta}_{k+1}$ is the matrix-vector operation $\boldsymbol{v} \to \boldsymbol{\Phi}\boldsymbol{v}$. Therefore, the required number of arithmetic operations to run p iterations of the CG update is comparable to that of a direct method based on the Cholesky factorization of $\boldsymbol{\Phi}$. Through an effective preconditioning strategy described in the next section, however, we can induce rapid convergence of CG for a typical precision matrix $\boldsymbol{\Phi}$ arising from the conditional distribution (1.3). In our numerical results, we indeed find that the distribution of $\boldsymbol{\beta}_k$ even for $k \ll p$ is indistinguishable from (1.3) for all practical purposes (Section 5.6).

It is worth emphasizing that our CG sampler does not require explicitly forming the precision matrix $\boldsymbol{\Phi}$ of the form (1.3) because the vector $\boldsymbol{\Phi}\boldsymbol{v}$ can be computed via operations $\boldsymbol{v} \to \boldsymbol{X}\boldsymbol{v}$ and $\boldsymbol{w} \to \boldsymbol{X}^\mathsf{T}\boldsymbol{w}$ along with simple element-wise multiplications. This is critical for our "large n and large p" applications for a couple of reasons. First, computing the term $\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}$ in $\boldsymbol{\Phi}$ often requires more computational efforts than a subsequent factorization of $\boldsymbol{\Phi}$ when $n > p$. Secondly, as mentioned earlier for a large sparse design matrix $\boldsymbol{X}$, storing an explicitly computed $\boldsymbol{\Phi}$ can significantly increase memory burden compared to storing $\boldsymbol{X}$ itself.

2.3. Convergence behavior of the CG method

When directly applied to a given linear system, the iterative solution $\{\boldsymbol{\beta}_k\}_{k=0,1,2,\ldots}$ of CG often displays slow convergence. Section 2.4 covers the topic of how to induce more rapid CG convergence for the system (2.5) arising from sparse Bayesian regression. In preparation, here we describe how the convergence behavior of CG is related to the structure of the positive definite matrix $\boldsymbol{\Phi}$.

Convergence behavior of CG can be partially explained by the following well-known error bound given in terms of the *condition number* $\kappa(\boldsymbol{\Phi})$, the ratio of the largest to the 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 k-th CG iterate $\boldsymbol{\beta}_k$ satisfies the following bound in its $\boldsymbol{\Phi}$-norm distance to the solution $\boldsymbol{\beta}$:*

$$\frac{\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq 2\left(\frac{\sqrt{\kappa(\boldsymbol{\Phi})} - 1}{\sqrt{\kappa(\boldsymbol{\Phi})} + 1}\right)^k. \tag{2.8}$$

See Trefethen and Bau (1997) for a proof. Theorem 2.2 guarantees fast convergence of the CG iterates when the condition number $\kappa(\boldsymbol{\Phi})$ is small. On the other hand, a large condition number does not always prevent rapid convergence of CG. This is because CG can also converge quickly 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 the 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 i.e. $\boldsymbol{\Phi} = \boldsymbol{F}\boldsymbol{F}^\mathsf{T} + \boldsymbol{I}$ for $\boldsymbol{F} \in \mathbb{R}^{p\times k}$.*

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

Theorem 2.2 and 2.3 are arguably the two most famous results on the convergence property of CG, perhaps because their conclusions are clear-cut and easy to understand. However, each theorem captures only an aspect of CG's convergence property and falls short of describing typical behavior in practical applications; the assumption behind Theorem 2.3 is unrealistic, while the bound of Theorem 2.2 is too pessimistic. To better capture behavior of CG in actual applications, we bring together the most useful of the known results that have been scattered around the numerical linear algebra literature and summarize them as the following rule of thumb. We make all the statements contained in the rule of thumb mathematically precise in Appendix B. As we will see, the rule of thumb points us to the design of an effective preconditioning strategy in the context of sparse Bayesian regression.

RULE OF THUMB 2.4. *Suppose that the eigenvalues $\nu_p(\boldsymbol{\Phi}) \leq \ldots \leq \nu_1(\boldsymbol{\Phi})$ of $\boldsymbol{\Phi}$ are clustered in the interval $[\nu_{p-s}, \nu_r]$ except for a small fraction of them. Then CG "knocks off" the outlying eigenvalues exponentially quickly and its convergence subsequently accelerates as if the effective condition number of $\boldsymbol{\Phi}$ is ν_r/ν_{p-s} rather than ν_1/ν_p in Eq 2.8. The r largest eigenvalues are effectively removed within r iterations, while the same number of smallest eigenvalues tends to delay the convergence longer.*

2.4. Preconditioning the CG method for rapid solutions of (2.5)

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

$$\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\beta}} = \tilde{\boldsymbol{b}} \quad \text{for } \tilde{\boldsymbol{\Phi}} = \boldsymbol{M}^{-1/2}\boldsymbol{\Phi}\boldsymbol{M}^{-1/2} \text{ and } \tilde{\boldsymbol{b}} = \boldsymbol{M}^{-1/2}\boldsymbol{b} \tag{2.9}$$

leads to faster convergence of the CG iterations. In practice, we only need $\boldsymbol{M}^{-1}$ and not $\boldsymbol{M}^{-1/2}$ since the algorithm can be implemented so that only the operation $\boldsymbol{v} \to \boldsymbol{M}^{-1}\boldsymbol{v}$ is required to solve the preconditioned system (2.9) 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. While a wide range of general techniques has been proposed, finding a good preconditioner to a given linear system remains "a combination of art and science." (Saad, 2003)

In light of Rule of Thumb 2.4, an effective preconditioner should modify the eigenvalue structure of $\boldsymbol{\Phi}$ so that the preconditioned matrix $\tilde{\boldsymbol{\Phi}}$ has more tightly clustered eigenvalues except perhaps for a small number of outlying eigenvalues. Larger outlying eigenvalues are preferable over smaller ones, as smaller ones cause more significant, if not severe, delays in the convergence of CG. In addition to the convergence rate, a choice of a preconditioner must take into account the one-time cost of computing the preconditioner $\boldsymbol{M}$ itself as well as the cost of operation $\boldsymbol{v} \to \boldsymbol{M}^{-1}\boldsymbol{v}$ during each CG iteration.

In the contexts of sparse Bayesian regression, the linear system (2.5) turns out to admit a deceptively simple yet highly effective preconditioner, obviating the need for more complex and computationally expensive preconditioning strategies. In fact, the choice

$$\boldsymbol{M} = \tau^{-2}\boldsymbol{\Lambda}^{-2} \tag{2.10}$$

yields a modified system (2.9) with an eigenvalue structure ideally suited to the CG application. With a slight abuse of terminology, we will call this matrix the *prior preconditioner* since it corresponds to the precision of $\boldsymbol{\beta} \,|\, \tau, \boldsymbol{\lambda}, \boldsymbol{\omega}$ ($\stackrel{d}{=} \boldsymbol{\beta} \,|\, \tau, \boldsymbol{\lambda}$) before observing $(\boldsymbol{y}, \boldsymbol{X})$. Note that this choice is different from the widely-used Jacobi preconditioner based on the diagonal elements of $\boldsymbol{\Phi}$; despite being a reasonable choice, the Jacobi preconditioner is substantially inferior to the prior preconditioner in typical applications (Section 3.3 and 5.4).

Our prior preconditioner can be motivated as follows. The shrinkage prior is employed either when 1) we want to impose a sparsity through an informative prior on the global shrinkage parameter τ or 2) we believe that the data support sparser models, say, in terms of the marginal likelihood. In either case, 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 $j = 1, \ldots, p$. (More precisely, we mean by $\tau\lambda_j \approx 0$ that the contribution of the term $x_{ij}\beta_j \,|\, \tau, \lambda_j$ is small in predicting the outcome.) This observation leads to the following simple heuristic. Note that the prior-preconditioned matrix is given by

$$\tilde{\boldsymbol{\Phi}} = \tau^2 \boldsymbol{\Lambda} \boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X} \boldsymbol{\Lambda} + \boldsymbol{I}_p \tag{2.11}$$

and the matrix $\tau^2 \boldsymbol{\Lambda} \boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X} \boldsymbol{\Lambda}$ has the (i, j)-th entry

$$\big(\tau^2 \boldsymbol{\Lambda} \boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X} \boldsymbol{\Lambda}\big)_{i,j} = (\tau\lambda_i)(\tau\lambda_j) \big(\boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X}\big)_{ij} \tag{2.12}$$

which is small when either $\tau\lambda_i \approx 0$ or $\tau\lambda_j \approx 0$. In other words, the entries of $\tau^2 \boldsymbol{\Lambda} \boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X} \boldsymbol{\Lambda}$ 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 (2.11) can be thought of as a perturbation of the identity with a matrix with approximate low-rank structure. As such, $\tilde{\boldsymbol{\Phi}}$ can be expected to have eigenvalues clustered around 1, except for a relatively small number of larger ones.

The above heuristic on the approximate low-rank structure of $\tilde{\boldsymbol{\Phi}}$ is formalized and quantified in Theorem 2.5. It is also worth noting, however, that it is too naive to conclude from the above heuristic that a good approximation to $\tilde{\boldsymbol{\Phi}}$ can be obtained by simply zeroing out $\tau\lambda_j$'s below some threshold. Such thresholding approach is successfully employed in Johndrow et al. (2018) for a related but different problem. In our context, however, numerical results clearly show that such approximation can be of a poor quality — see the supplement Section S4.

THEOREM 2.5. *Let $\lambda_{(k)} = \lambda_{j_k}$ denote the k-th largest element of $\{\lambda_1, \ldots, \lambda_p\}$. The eigenvalues of the prior-preconditioned matrix* (2.11) *then satisfies*

$$1 \le \nu_{k+1}(\tilde{\boldsymbol{\Phi}}) \le 1 + \tau^2 \lambda^2_{(k+1)} \, \nu_1\big(\boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X}\big) \tag{2.13}$$

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

$$1 \le \nu_{k+\ell}(\tilde{\boldsymbol{\Phi}}) \le 1 + \tau^2 \lambda^2_{(k+1)} \, \nu_{\ell+1}\big((\boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X})_{(-k)}\big) \le 1 + \tau^2 \lambda^2_{(k+1)} \, \nu_{\ell+1}\big(\boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X}\big) \tag{2.14}$$

for any $k, \ell \geq 0$ *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 the CG — when most of $\tau\lambda_j$'s are close to zero. Through its dependence on $\nu_\ell\big(\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}\big)$, the bound (2.14) further shows that rapid CG convergence is also expected when the eigenvalues of $\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}$ decays quickly, which tends to happen when there is high-collinearity among the predictors.

Theorem 2.5 can also be used to directly relate the CG approximation error under the prior preconditioner to the decay rate in $\tau\lambda_{(k)}$'s:

THEOREM 2.6. *The prior-preconditioned CG applied to* (2.5) *yields iterates satisfying the following bound for any* $m, m' \geq 0$*:*

$$\frac{\|\boldsymbol{\beta}_{m+m'} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq 2\left(\frac{\widetilde{\kappa}_m^{1/2} - 1}{\widetilde{\kappa}_m^{1/2} + 1}\right)^{m'} \quad \text{where} \quad \widetilde{\kappa}_m = 1 + \min_{k+\ell=m} \tau^2\lambda_{(k+1)}^2\nu_{\ell+1}\big((\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X})_{(-k)}\big)\,. \tag{2.15}$$

See Appendix A for proofs of Theorem 2.5 and 2.6.

To illustrate the implication of Theorem 2.6 in concrete terms, suppose that a posterior draw $\tau, \boldsymbol{\lambda}, \boldsymbol{\omega}$ satisfies $\tau^2\lambda_{(m+1)}^2\nu_1\big(\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}\big) \leq 100$ for some m. In this case, we have $\log_{10}\big(\widetilde{\kappa}_m^{1/2} - 1\big)\big/\big(\widetilde{\kappa}_m^{1/2} + 1\big) \leq -0.086$. So the bound (2.15) implies

$$\frac{\|\boldsymbol{\beta}_{m+m'} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq 2 \cdot 10^{-0.086m'}. \tag{2.16}$$

For instance, after $m + 100$ iterations, 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 introduced the prior-preconditioning strategy ultimately for the purpose of efficient posterior computation under sparse regression models. All the results have so far been formulated in linear algebraic languages, however. To provide a useful guideline on the performance of the CG-accelerated Gibbs sampler in practical applications, we now summarize the above discussions in a more statistical language.

RULE OF THUMB 2.7. *The prior preconditioner* (2.10) *induces rapid convergence of the CG applied to the linear system* (2.5) *when the posterior of* $\boldsymbol{\beta}$ *concentrates on sparse vectors. (With continuous shrinkage priors, by sparsity we mean that most of* β_j*'s are virtually zero in their magnitudes.) As the sparsity of* $\boldsymbol{\beta}$ *increases, the average convergence rate of the CG sampler also increases.*

The statements above are born out by an illustrative example of Section 3 using synthetic sparse regression posteriors. Also, 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, Theorem 2.5, and Theorem 2.6).

2.5. General principle behind prior-preconditioning approach

We close our discussion of preconditioning techniques by providing alternative heuristics behind the prior preconditioner. While not as quantitative as Theorem 2.5, the following general principle suggests that the prior-preconditioning is effective beyond sparse regression settings for any Bayesian computation involving conditionally Gaussian distributions. In the context of the CG sampler, the preconditioned matrix $\boldsymbol{M}^{-1/2}\boldsymbol{\Phi}\boldsymbol{M}^{-1/2}$ represents the precision matrix of the transformed parameter $\tilde{\boldsymbol{\beta}} = \boldsymbol{M}^{1/2}\boldsymbol{\beta}$. In fact, preconditioning the linear system (2.5) with a preconditioner $\boldsymbol{M}$ is equivalent to applying a parameter transformation $\boldsymbol{\beta} \to \boldsymbol{M}^{1/2}\boldsymbol{\beta}$ before employing the CG sampler. That is, we can apply one of the two strategies — precondition the linear system or apply the parameter transformation — to achieve exactly the same effect on the speed of the CG sampler.

When we choose the prior precision as the preconditioner, the transformed parameter $\boldsymbol{M}^{1/2}\boldsymbol{\beta}$ a priori has the identity precision matrix, before its distribution is modified via the likelihood. This perspective, combined with the fact that the eigenvalues of $\tilde{\boldsymbol{\Phi}}$ represents the posterior precisions of $\boldsymbol{M}^{1/2}\boldsymbol{\beta}$ along its principal components, suggests the following principle:

Principle Behind Prior-preconditioning 2.8. *Under a strongly informative prior, the posterior looks like the prior except in a small number of directions along which the data provide significant information. This translates into the eigenvalues of the prior-preconditioned matrix $\tilde{\boldsymbol{\Phi}}$ clustering around 1 except for a relatively small number of large eigenvalues.*

The eigenvalue structure of the prior-preconditioned matrix $\tilde{\boldsymbol{\Phi}}$ as predicted above is indeed observed across all of our numerical examples — see Figure 3.2, S3, and S6.

3. Demonstration of the CG sampler performance on simulated data

In addition to the prior preconditioning strategy introduced in Section 2.4, there remain a few more important details to successfully apply the CG sampler to sparse regression problems in practice. Preconditioning is undoubtedly the most essential ingredient of the CG sampler, however, and we defer other practical details to Section 4. Instead, we now turn to demonstrating the performance of the CG sampler applied to distributions of the form (1.3) as they arise from actual posterior distributions of sparse Bayesian logistic regression models. We illustrate how the CG convergence rates are affected by different preconditioning strategies and by corresponding eigenvalue distributions of the preconditioned matrices. Also, we use simulated data with varying numbers of non-zero 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.

3.1. Choice of shrinkage prior: Bayesian bridge

While a variety of global-local shrinkage priors of the form (1.1) are available, we adopt the Bayesian bridge prior of Polson et al. (2014) in our implementation of

the CG-accelerated Gibbs sampler. We make this choice for the following reasons. First, the Gibbs sampler under the Bayesian bridge tends to demonstrate better mixing in the global shrinkage parameter τ. This is because the Bayesian bridge formulation allows the update of τ from the distribution of $\tau \,|\, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{y}, \boldsymbol{X}$ with $\boldsymbol{\lambda}$ marginalized out (Polson et al., 2014). Secondly, many of the alternative shrinkage priors have extremely heavy tails that can be problematic in the logistic regression context. Under such heavy-tailed priors, the posterior ends up having heavy-tails when the parameters are only weakly identified from the data, causing issues both in terms of posterior computation and inference (Ghosh et al., 2018; Piironen and Vehtari, 2017).

Under the Bayesian bridge, the local shrinkage parameter λ_j's are given independent alpha-stable distributions with index of stability $\alpha/2$ with $0 < \alpha \leq 1$. This choice induces to a prior on $\beta_j \,|\, \tau$, when λ_j is marginalized out, such that

$$\pi(\beta_j \,|\, \tau) \propto \tau^{-1} \exp\left(-|\beta_j/\tau|^{\alpha}\right). \tag{3.17}$$

The case $\alpha = 1$ coincides with the Bayesian lasso. The distribution of $\beta_j \,|\, \tau$ becomes "spikier" as $\alpha \to 0$, placing greater mass around 0 while at the same time having heavier tails. While an alpha-stable distribution has no closed-form expression, there are algorithms available to efficiently sample from the posterior distribution of $\lambda_j \,|\, \beta_j, \tau$ (Polson et al., 2014). In typical applications, the marginal likelihood of data favors the values $\alpha < 1$ but only very weakly identifies it (Polson et al., 2014), so in our implementation we simply set $\alpha = 1/2$.

3.2. Experimental set-up

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

$$\boldsymbol{x}_i = \sum_{\ell=1}^{99} f_{i,\ell} \boldsymbol{u}_\ell + \boldsymbol{\epsilon}_i \;\text{ for } f_{i,\ell} \sim \mathcal{N}\left(0, (100-\ell)^2\right) \text{ and } \boldsymbol{\epsilon}_i \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}_p\right). \tag{3.18}$$

This is equivalent to independently sampling $\boldsymbol{x}_i \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\intercal}\big)$, where $\boldsymbol{D}$ is a diagonal matrix with $\sqrt{D_{\ell\ell}} = \max\{100 - \ell, 1\}$ and $\boldsymbol{U}$ is an orthonormal matrix sampled uniformly from the space of orthonormal matrices. We then center and standardize the predictors as is commonly done in sparse regression models (Hastie et al., 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 non-zero regression coefficients. More specifically, we consider a sparse regression coefficient $\boldsymbol{\beta}_{\text{true}}$ with $\beta_{\text{true},\,j} = \mathbb{1}\{j \leq K\}$ for each $K = 10, 20$, and 50. In all

three scenarios, the binary outcome y_i's are generated from the logistic model as $y_i \,|\, \boldsymbol{\beta}_{\text{true}}, \boldsymbol{x}_i \sim \text{Ber}(p_i)$ for $\text{logit}(p_i) = \boldsymbol{x}_i^\intercal \boldsymbol{\beta}_{\text{true}}$.

For each of the simulated data sets, we obtain a posterior sample of $\boldsymbol{\omega}, \tau, \boldsymbol{\lambda} \,|\, \boldsymbol{y}, \boldsymbol{X}$ by running the current state-of-the-art Polya-Gamma augmented Gibbs sampler using the direct linear algebra to sample $\boldsymbol{\beta}$ from its conditional distribution (1.3). We confirm the convergence of the Markov chain through the traceplot of the posterior log density of $\boldsymbol{\beta}, \tau \,|\, \boldsymbol{y}, \boldsymbol{X}$ (with $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ marginalized out). We can consider the state-of-the-art Gibbs sampler "exact" in a sense that the direct linear algebra has no potential convergence issues of the CG method. (It is, however, still affected by errors from finite precision arithmetic as always.)

Having obtained a posterior sample $\boldsymbol{\omega}, \tau, \boldsymbol{\lambda}$, we sample a vector $\boldsymbol{b}$ as in (2.4) and apply CG to the linear system (2.5). We then compare the CG iterates $\{\boldsymbol{\beta}_k\}_{k \geq 0}$ to the exact solution $\boldsymbol{\beta}_{\text{direct}}$ obtained by solving the same system with the direct linear method based on the Cholesky factorization. We repeat this process for 8 random replications of the right-hand vector $\boldsymbol{b}$ for the linear system (2.5).

3.3. Results: convergence rates, eigenvalues, and sparsity of coefficients

Figure 3.1 shows the CG approximation error as a function of the number of CG iterations, whose cost as we discuss in Section 2.2 is dominated by matrix-vector operations of the form $\boldsymbol{v} \to \boldsymbol{\Phi}\boldsymbol{v}$. Here we characterize the error as the average of the relative error $|(\boldsymbol{\beta}_k - \boldsymbol{\beta}_{\text{direct}})_j / (\boldsymbol{\beta}_{\text{direct}})_j|$ across all the coefficients. For each line on the plot, this error is averaged in the log-scale over the 8 random replications of the right-hand vector $\boldsymbol{b}$. Plots in the supplement Section S1 show, however, that the CG convergence behavior remains qualitatively similar regardless of choice of the error metric and varies little across the different right-hand vectors. In particular, the superior convergence rate of the prior-preconditioning over the Jacobi one holds consistently.

First, we focus on the results corresponding to the prior-preconditioned CG, 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 error, say, in estimating $\mathbb{E}[\boldsymbol{\beta} \,|\, \boldsymbol{y}, \boldsymbol{X}]$. In fact, with a relatively small number of additional CG iterations, the distance reaches the level of machine precision manifested as the "plateaus" of the approximation error seen in the blue dash-dot and orange dashed lines with circles.

Along with the approximation errors based on the prior preconditioner, as a benchmark we also compute the errors based on the Jacobi preconditioner $\boldsymbol{M} = \text{diag}(\Phi_{11}, \ldots, \Phi_{pp})$. The Jacobi preconditioner is the simplest general-purpose preconditioner and usually performs well for linear systems like ours when the diagonals of $\boldsymbol{\Phi}$ is significantly larger than the off-diagonals (Golub and Van Loan, 2012). In the context of the CG sampler, the Jacobi preconditioning coincides with the use of the conditional precisions $\beta_j \,|\, \ldots, \boldsymbol{\beta}_{-j}$ as a preconditioner. While more complex general-purpose preconditioners exist, they require substantially more computational efforts to compute them before the CG iterations can even get started (Golub and Van Loan, 2012). We therefore do not consider those preconditioners

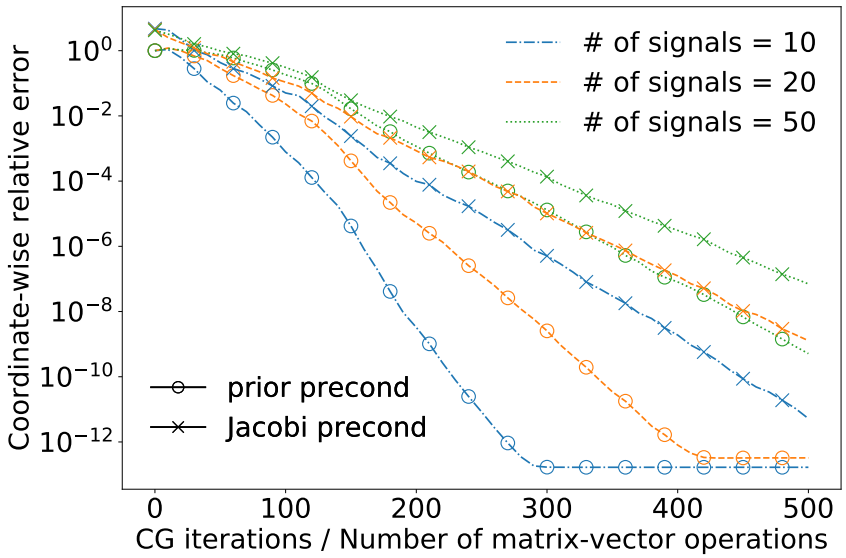
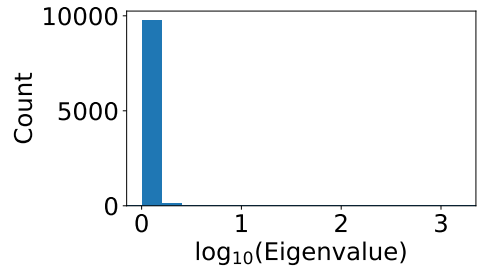


Fig. 3.1. Plot of the CG approximation error as a function of the number of CG iterations when the CG sampler is applied to synthetic sparse regression posteriors with varying number of signals.

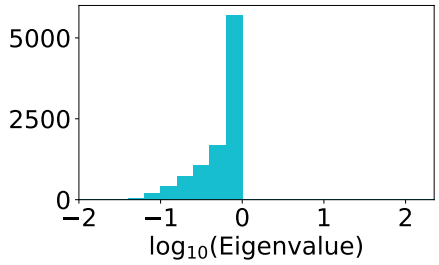
here.

It is evident from Figure 3.1 that the prior preconditioner induces more rapid CG convergence than the Jacobi preconditioner for the purpose of our CG-accelerated Gibbs sampler. The difference in convergence speed is more pronounced with sparser true regression coefficients. Studying the eigenvalue distributions of the respective preconditioned matrices provides further insight into the observed convergence behaviors. Figure 3.2 (a) & (b) show the eigenvalue distributions of the preconditioned matrices based on a posterior sample from the synthetic data with 10 non-zero coefficients. The trimmed version of the histograms are also given to 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 we developed in Section 2.4. 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).

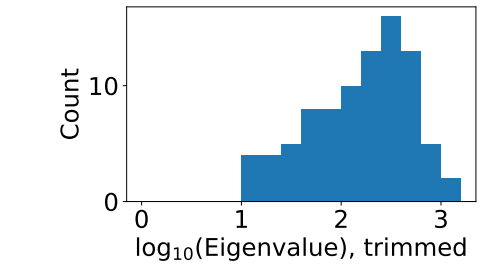
Finally, we turn our attention to the relationship as seen in Figure 3.1 between the CG convergence rate and the sparsity in the underlying true regression coefficients. The CG convergence is clearly quicker when the true regression coefficients are sparser. To understand this relationship, it is informative to look at the values of $\tau\lambda_j = \mathbb{E}[\beta_j \,|\, \tau, \boldsymbol{\lambda}]$ drawn from the respective posterior distributions. Figure 3.3 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 $K = 10$ and $K = 50$, shown on the left and



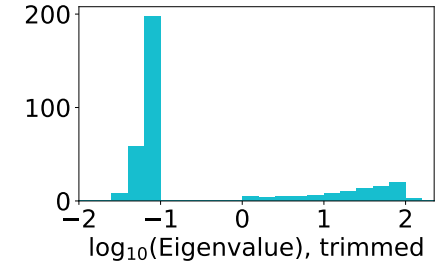
(a) Based on the prior preconditioner & synthetic data with 10 non-zero coefficients.



(b) Based on the Jacobi preconditioner & synthetic data with 10 non-zero coefficients.



(c) Based on the prior preconditioner & synthetic data with 50 non-zero coefficients.



(d) Based on the Jacobi preconditioner & synthetic data with 50 non-zero coefficients.

Fig. 3.2. Histograms of the eigenvalues of the preconditioned matrices. The preconditioned matrices are based on a posterior sample from the synthetic data. The trimmed versions are shown on the lower row, 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 preconditioner. 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. Note however that the axis ranges are not kept constant.

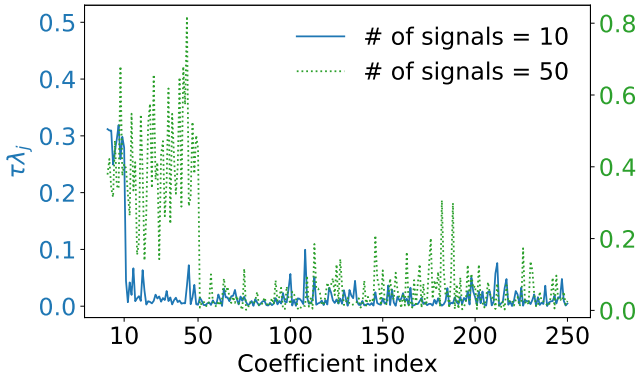


Fig. 3.3. Plot of the posterior samples of $\tau\lambda_j$'s for $j = 1, \ldots, 250$. The two lines correspond to the two distinct posteriors; the data sets are simulated with true regression coefficients having nonzero coefficients $\beta_{\text{true},j} = 1$ for $j \leq 10$ and $j \leq 50$.

right respectively, to facilitate the 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 $\tau\lambda_j$ away from zero. These relatively large $\tau\lambda_j$'s contribute to the delayed convergence of CG as quantified in 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 3.3, one can see that $\tau\lambda_1, \ldots, \tau\lambda_K$ corresponding to the true signals are not as well separated from the rest of $\tau\lambda_j$'s when $K = 50$. In fact, the histograms on the left of Figure 3.4 shows that the distribution of $\tau\lambda_j$'s for $K = 50$ are shifted toward larger values compared to that for $K = 10$. 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 $K = 10$ case while $\tau \approx 6.7 \times 10^{-3}$ for the $K = 50$ 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 3.4 show the distribution of the 250 largest $\tau\lambda_j$'s after scaling them by $\max_j \tau\lambda_j$. The figure makes it clear that $\tau\lambda_j$'s corresponding to the true signals are much more well separated from the rest when $K = 10$. Overall, the slower decay in the largest values of $\tau\lambda_j$'s when $K = 50$ results in the eigenvalues of the prior preconditioner having a less tight cluster around 1; compare the eigenvalue distributions of Figure 3.2.(a) & (b) to those of Figure 3.2.(c) & (d).

4. Practical CG sampler details for sparse regression

4.1. Initial vector for CG iterations

As suggested by Theorem 2.2 and by more thorough analysis of the CG convergence behavior in Appendix B, generally speaking the CG iterations decrease the distance between the iterates $\boldsymbol{\beta}_k$'s and the exact solution $\boldsymbol{\beta}$ relative to the initial error $\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}$. Hence there is a benefit to choosing the initial vector $\boldsymbol{\beta}_0$ with a small initial error $\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}$. It should be noted, however, that choice of a preconditioner determines the exponential convergence rate of CG and, comparatively, choice of

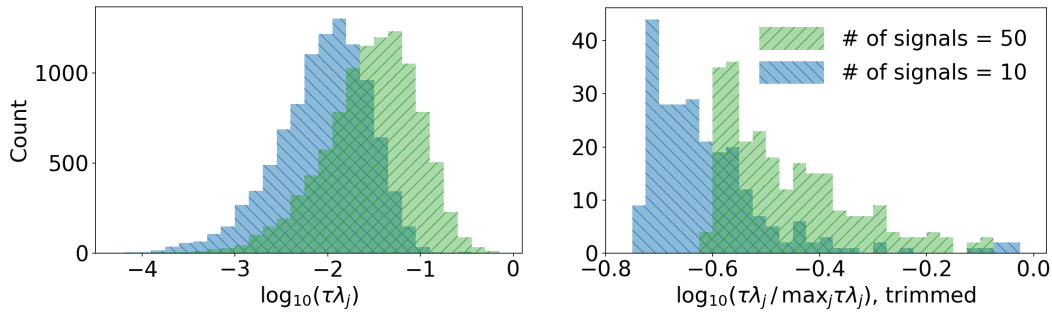


Fig. 3.4. Histograms of the posterior samples of $\tau\lambda_j$'s. The histograms with two different colors correspond to the distinct posteriors with 10 and 50 non-zero regression coefficients. The histograms on the right better expose the relative tail behaviors in the two distributions of $\tau\lambda_j$'s by taking only the 250 largest values and plotting their magnitude relative to $\max_j \tau\lambda_j$.

an initial vector often has a smaller effect on the approximation error. In other words, once the initial vector is chosen within a reasonable range, we should not expect a dramatic gain from further fine-tuning. In fact, when sampling $\boldsymbol{\beta}$ from a sparse regression posterior, we find it difficult to improve much over a simple initialization $\boldsymbol{\beta}_0 = \mathbf{0}$, which is a reasonable choice as most coefficients are shrunken to near zero. Only small ($< 10\%$) though consistent improvements were observed by one of the other approaches with which we experimented — see the supplement Section S2.1.

4.2. Termination criteria for CG method

An iterative method must be supplied with a termination criteria to decide when the current iterate $\boldsymbol{\beta}_k$ is close enough to the exact solution. A CG termination criteria used in most of the existing linear algebra libraries is based on the ℓ^2 norm of the residual $\boldsymbol{r}_k = \boldsymbol{\Phi}\boldsymbol{\beta}_k - \boldsymbol{b}$. This is mostly for convenience reasons as the residual norm is easily computed as a bi-product of the CG iterations. At least, the residual norm can be related to $\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_2$ as

$$\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_2 = \|\boldsymbol{\Phi}^{-1}\boldsymbol{r}_k\|_2 \leq \|\boldsymbol{\Phi}^{-1}\|_2 \|\boldsymbol{r}_k\|_2. \tag{4.19}$$

For the purpose of sampling a Gaussian vector $\boldsymbol{\beta}$, however, it is not at all clear when $\|\boldsymbol{r}_k\|_2$ or $\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_2$ can be considered small enough. To address this problem, we develop an alternative metric tailored toward our CG sampler for sparse regression.

For our CG sampler, we propose to assess the CG convergence in terms of the ℓ^2 norm of the prior-preconditioned residual $\tilde{\boldsymbol{r}}_k = \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{b}} = \tau\boldsymbol{\lambda} \odot \boldsymbol{r}_k$, where $\odot$ denotes the element-wise multiplication of two vectors. More specifically, we use the termination criteria

$$p^{-1/2}\|\tilde{\boldsymbol{r}}_k\|_2 = \left\{p^{-1}\textstyle\sum_{j=1}^{p}(\tilde{\boldsymbol{r}}_k)_j^2\right\}^{1/2} \leq 10^{-6}, \tag{4.20}$$

in terms of the root-mean-squared residual $p^{-1/2}\|\tilde{\boldsymbol{r}}_k\|_2$.

In the supplement Section S2.2, we explain the utility of the norm $\|\tilde{\boldsymbol{r}}_k\|_2$ as an approximate upper bound to the following quantity:

$$\left\|\boldsymbol{\xi}^{-1} \odot (\boldsymbol{\beta}_k - \boldsymbol{\beta})\right\|_2 \quad \text{with} \quad \xi_j^2 = \mathbb{E}\big[\,\beta_j^2 \,|\, \boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X}\big]. \tag{4.21}$$

The standardization by second moment ensures that, when the computed error is small, all the coordinates of $\boldsymbol{\beta}_k$ are close to those of $\boldsymbol{\beta}$ either in terms of their means or variances of the target Gaussian distribution. In all our examples, we find this metric to work very well in quantifying the numerical error for the purpose of the CG-accelerated sampling; see Section 5.4 and 5.6 for illustrations.

4.3. Incorporating intercept and predictors with uninformative prior

When fitting a sparse regression model, standard practice is to include an intercept β_0 without any shrinkage, often with the improper flat prior $\pi(\beta_0) \propto 1$ (Park and Casella, 2008). Additionally, there may be predictors of particular interests, inference for whose regression coefficients is more appropriately carried out with uninformative or weakly-informative priors without shrinkage; see Zucknick et al. (2015) as well as the application in Section 5 for examples of such predictors. Our CG-accelerated Gibbs sampler can accommodate such predictors with an appropriate modification to the prior preconditioner proposed above.

For notational convenience, suppose that the regression coefficients are indexed so that the first $(q+1)$-th coefficients $\beta_0, \beta_1, \ldots, \beta_q$ are to be estimated without shrinkage. We further assume that the unshrunk coefficients are given independent Gaussian priors $\beta_j \sim \mathcal{N}(0, \sigma_j^2)$ for $0 < \sigma_j \leq \infty$ where $\sigma_j = \infty$ denotes an improper prior $\pi(\beta_j) \propto 1$. The precision matrix of $\boldsymbol{\beta} \,|\, \boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X}$ then is given by

$$\boldsymbol{\Phi} = \boldsymbol{X}^\intercal \boldsymbol{\Omega} \boldsymbol{X} + \begin{bmatrix} \mathrm{diag}(\boldsymbol{\sigma})^{-2} & \boldsymbol{0} \\ \boldsymbol{0} & \tau^{-2}\boldsymbol{\Lambda}^{-2} \end{bmatrix} \tag{4.22}$$

for $\boldsymbol{\sigma} = (\sigma_0, \ldots, \sigma_q)$ where we employ the convention $1/\sigma_j = 0$ if $\sigma_j = \infty$. From the computational point of view, the unshrunk coefficients $\beta_0, \ldots, \beta_q$ are distinguished from the shrunk ones by the fact that their prior scales σ_j (before conditioning on $\boldsymbol{y}$ and $\boldsymbol{X}$) typically have little to do with their posterior scales (after conditioning on $\boldsymbol{y}$ and $\boldsymbol{X}$). For this reason, a naively modified preconditioner $\boldsymbol{M} = \mathrm{diag}(\boldsymbol{\sigma}^{-2}, \tau^{-2}\boldsymbol{\lambda}^{-2})$ may not be appropriate, especially for coefficients with $\sigma_j \gg 1$ corresponding to uninformative priors.

We propose a modified preconditioner of the form $\boldsymbol{M} = \mathrm{diag}(\boldsymbol{\gamma}^{-2}, \tau^{-2}\boldsymbol{\lambda}^{-2})$ for appropriately chosen $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots, \gamma_q)$. Under the proposed form of a modified preconditioner, it can be shown that all but $q+1$ eigenvalues of the preconditioned matrix $\tilde{\boldsymbol{\Phi}}$ are "well-behaved" for the purpose of rapid CG convergence. Our goal, therefore, is to choose γ_j's to control the behavior of the $q+1$ additional eigenvalues introduced by the unshrunk coefficients. The detailed analysis of how to achieve this goal is somewhat involved and is provided in the supplement Section S2.3, but in the end we show that

$$\gamma_j = c\,\widehat{\eta}_j \quad \text{for} \quad \widehat{\eta}_j^2 \approx \mathrm{var}(\beta_j \,|\, \boldsymbol{y}, \boldsymbol{X}) \tag{4.23}$$

with some $c \geq 1$ is a good choice. The factor $c \geq 1$ is included since, as explained in Section S2.3, it is better to err on the side of choosing γ_j's larger than smaller. In our numerical results, we use $c = 2$ and estimate $\widehat{\eta}_j$ from earlier MCMC iterations.

While we find a poor choice of γ_j's to significantly delay CG convergence in practice, we also point out the following: once γ_j's are chosen within reasonable ranges, their precise values have rather small effect on the convergence rate when q is small. This is because, under the proposed modified preconditioner, all but $(q+1)$ eigenvalues are well-behaved regardless of the choice of $\gamma_0, \ldots, \gamma_q$. As CG has an ability to eventually "knock off" the extreme eigenvalues (Rule of Thumb 2.4), the additional $q + 1$ eigenvalues will generally have limited effects in its convergence behavior.

5. Application: comparison of two drugs using healthcare databases

In this section, we demonstrate how CG-acceleration delivers an order of magnitude speed-up in the posterior computation for a large-scale observational study. We apply sparse Bayesian logistic regression to conduct a comparative study of two anti-coagulants *dabigatran* and *warfarin*. These drugs are administered to reduce the risk of blood clot formation, but as a side effect can increase the incidence rates of bleeding. The goal of the study is to quantify which of the two drugs have a lower risk of intracranial hemorrhage (bleeding inside the skull). This question has previously been investigated by Graham et al. (2015) and our analysis yields clinical findings consistent with theirs (see Section 5.7).

We are particularly interested in sparse Bayesian regression as a tool for the Observational Health Data Sciences and Informatics (OHDSI) collaborative (Hripcsak et al., 2015). Therefore, we follow the OHDSI protocol in pre-processing of the data as well as in the treatment effect estimation procedure. In particular, sparse regression plays a critical role in eliminating the need to hand-pick confounding factors; this enables the application of a reproducible and consistent statistical estimation procedure to tens of thousands of observational studies (Schuemie et al., 2018b,a; Tian et al., 2018).

5.1. Data set

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 were new 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, pre-existing 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 at `http://www.ohdsi.org/web/atlas/#/cohortdefinition/{2978,2979,2981}`.

Each patient has or is exposed to only a small subsets of all the possible pre-existing conditions, treatments, and drugs. The design matrix $\boldsymbol{X}$ therefore is sparse, with only 5% of the entries being non-zero. (The density of $\boldsymbol{X}$ would have been 1% without the screening of the infrequent predictors.) Another noteworthy feature of the data is the low incidence rates of intracranial hemorrhage, so that the outcome indicator $\boldsymbol{y}$ has non-zero entries $y_i = 1$ for only 192 out of 72,489 patients.

5.2. Statistical approach: treatment effect estimation via sparse regression

To estimate the effect of treatment by dabigatran over warfarin on the outcome of interest, we use a doubly-robust method for average treatment effect estimation with propensity score stratification. The actual procedure and essential ideas are described below, but we refer the readers to Stuart (2010) and the references therein for further details.

Within the framework of propensity score methods, the treatment effect estimation proceeds in two stages. First, the *propensity score* $\mathbb{P}(\, T_i = 1 \,|\, \boldsymbol{x}_i)$ of the treatment assignment to dabigatran of the i-th individual is estimated by the logistic model

$$\text{logit}\{\mathbb{P}(\, T_i = 1 \,|\, \boldsymbol{x}_i)\} = \beta_0 + \boldsymbol{x}_i^\intercal \boldsymbol{\beta}. \tag{5.24}$$

The quantiles of the estimated propensity scores are then used to stratify the population into subgroups of equal sizes. Following a typical recommendation, we choose the number of subgroups 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 ($T_i = 0$) groups.

After the propensity score stratification, we estimate the average treatment effect α_0 via the logistic model

$$\text{logit}\{\mathbb{P}(\, y_i = 1 \,|\, \boldsymbol{\alpha}, \boldsymbol{\beta}, s_i, \boldsymbol{x}_i)\} = \beta_0 + \alpha_0 T_i + \sum_{m=2}^{M} \alpha_m \mathbb{1}\{s_i = m\} + \boldsymbol{x}_i^\intercal \boldsymbol{\beta}, \tag{5.25}$$

where a categorical variable s_i denotes the strata membership of the i-th individual. (With a slight abuse of notation, we use $\boldsymbol{\beta}$ in both of the models (5.24) and (5.25) to denote the regression coefficients of predictors.) The indicator of $s_i = 1$ is excluded from the model for identifiability. The inclusion of the predictor $\boldsymbol{x}_i$ in the above model is technically not necessary if the distributions of predictors are perfectly balanced between the treatment and control groups within each strata $s_i = m$. Controlling for the predictor $\boldsymbol{x}_i$, however, makes the treatment estimation procedure more robust to potential misspecification in the propensity score model (5.24) as well as to any predictor imbalances that remain after conditioning on s_i.

Each of the regression models (5.24) and (5.25) involves a large number of features, making reliable estimation virtually impossible without some regularization or sparsity assumption. Sparse Bayesian regression is one promising approach, with an opportunity for future extensions such as hierarchical modeling across different hospitals.

5.3. Prior choice and posterior computation

We fit the models (5.24) and (5.25) using the Bayesian bridge shrinkage prior (see Section 3.1) on the regression coefficients. We place a reference prior $\pi(\tau) \propto 1/\tau$ on the global shrinkage parameter (Berger et al., 2009). An uninformative prior $\pi(\beta_0) \propto 1$ is used for the intercept. For the average treatment effect and propensity score strata effects, we place weakly informative $\mathcal{N}(0, 1)$ priors.

For posterior inference, we implemented two Gibbs samplers that differ only in their methods for the conditional updates of $\boldsymbol{\beta}$ from the distribution (1.3). One of the samplers uses the proposed CG sampler, while the other uses a traditional direct method based on the Cholesky factorization. We refer to the respective sampler as the *CG-accelerated* and *direct* Gibbs sampler. The other conditional updates follow the approaches described in Polson et al. (2014).

5.4. Overall speed-up from CG acceleration and posterior characteristics

We implement the Gibbs samplers in Python and run on 2015 iMac with Intel Core i7 processors. We run the Markov chains for 1,500 and 2,000 iterations for the propensity score and treatment effect model respectively, yielding 1,000 post-convergence samples. In total, the direct Gibbs sampler requires 77.4 hours for the propensity score model and 107 hours for the treatment effect model. On the other hand, the CG-accelerated sampler finishes in 7.04 and 4.36 hours, yielding **11-fold** and **25-fold** speed-ups.

We can explain the difference in the magnitudes of CG-acceleration between the two models in terms of the posterior sparsity structures of the regression coefficients (Section 2.4 and 3.3). For this purpose, we now examine the posteriors focusing only on the sparsity structure; other scientifically important aspects of the posteriors are discussed in Section 5.7. As a measure of sparsity, for each regression coefficient we consider one-sided tail probability, the larger of the posterior probabilities of $\beta_j > 0$ or $\beta_j < 0$, as well as the magnitude of posterior means.

For the propensity score model, 40 and 85 out of the 22,175 regression coefficients have one-sided tail probabilities above 97.5% and 90% respectively. Only 79 regression coefficients have the magnitude of their posterior means above 0.1, while 18,161 (81.9%) of the coefficients have the magnitude below 0.01.

For the doubly-robust treatment effect model, some of the regression coefficients have bi-modal marginal posterior distributions and their samples occasionally deviate away from zero. Averaged over all the posterior draws, however, except for the fixed effects no regression coefficient comes out as significantly different from 0. This suggests that the propensity score stratification is indeed successful in balancing the distribution of $\boldsymbol{x}_i$'s and thus it may have been unnecessary to include those predictors in the treatment effect model. Of course, we can only conclude this after actually fitting the doubly-robust model. As the Gibbs sampler requires close to 1,000 iterations before convergence, the burn-in iterations alone would be a significant computational burden if it were not for the CG-acceleration.

5.5. Mechanism behind CG acceleration

For both Gibbs samplers, the conditional updates of $\boldsymbol{\beta}$ from (1.3) dominate the computational times. To better understand the mechanism behind the CG-acceleration, therefore, we only need to examine the convergence rates of the CG sampler at each Gibbs iteration. Here we focus on a Gibbs update of $\boldsymbol{\beta}$ after the burn-in iterations for the propensity score model (5.24). Section S3 in the supplement provides additional analysis along with the corresponding results for the doubly-robust treatment effect model (5.25).

As done in Section 3, we examine how quickly the CG method finds an accurate solution to the linear system (2.5). The CG iterates $\boldsymbol{\beta}_k$ are compared against the exact solution $\boldsymbol{\beta}_{\text{direct}}$ found by a direct linear algebra via the Cholesky decomposition. Figure 5.5 shows the distances between $\boldsymbol{\beta}_k$ and $\boldsymbol{\beta}_{\text{direct}}$ as a function of the number of CG iterations k.

We first look at the solid blue line which tracks the root mean squared residual $p^{-1/2}\|\tilde{\boldsymbol{r}}_k\|_2$ as introduced in Section 4.2. The dotted vertical line indicates when the magnitude of the prior-preconditioned residual $\tilde{\boldsymbol{r}}_k$ falls below the termination criteria of (4.20). The termination occurs at the $k = 133$ iterations and the CG sampler consequently spends only $9\% \approx 1/11$ of the computational time relative to the direct Gibbs update. Analysis of Section 5.6 confirms that the CG approximation error at termination is so small that it does not affect the stationary distribution of the Gibbs sampler in any significant way. The number of iterations at termination fluctuates from one iteration to another of the Gibbs sampler since the linear system (2.5) depends on the random quantities $\boldsymbol{\omega}$, τ, $\boldsymbol{\lambda}$, and $\boldsymbol{b}$. This fluctuation is rather small, however. After the Gibbs sampler has converged, we rarely observe a deviation of more than $5 \sim 10\%$ from the average number of iterations — see the supplement Section S3.

In Section 4.2, we argue that $\|\tilde{\boldsymbol{r}}_k\|_2$ is a surrogate and approximate upper-bound for a more easily interpretable error metric. This is empirically confirmed here, by comparing the solid to dashed blue line; the dashed one tracks the root mean second-moment normalized error

$$\left\{p^{-1}\textstyle\sum_j \hat{\xi}_j^{-2}(\boldsymbol{\beta}_k - \boldsymbol{\beta}_{\text{direct}})_j^2\right\}^{1/2} \quad \text{with} \quad \hat{\xi}_j^2 \approx \mathbb{E}[\beta_j^2 \,|\, \boldsymbol{y}, \boldsymbol{X}] \tag{5.26}$$

which is computed as a proxy for (4.21). The standardization by second moment ensures that, when the computed error is small, all the coordinates of $\boldsymbol{\beta}_k$ are close to those of $\boldsymbol{\beta}_{\text{direct}}$ either in terms of their posterior means or variances.

Finally, we compare the blue and orange dashed lines to assess the relative CG convergence rates under the prior and Jacobi preconditioner. It is clear that the advantage of the prior preconditioner, as demonstrated in the simulated examples of Section 3, continues to hold in this real data example. The observed convergence behaviors under the two preconditioners are again well explained by the eigenvalue distributions of the respective preconditioned matrices — see Figure S3 in the supplement.

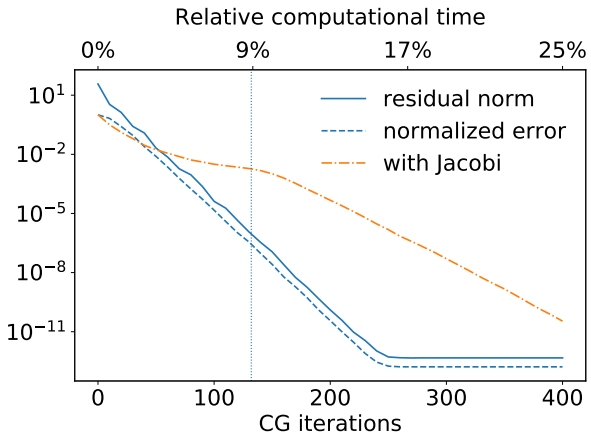


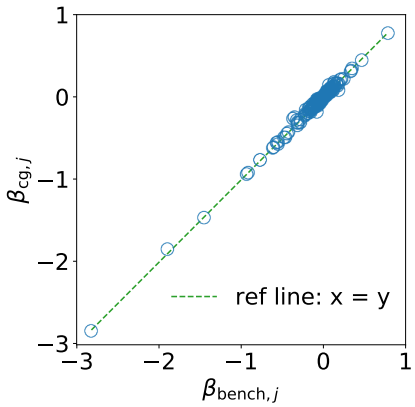
Fig. 5.5. Plot of the CG errors during a conditional update of $\boldsymbol{\beta}$ within the propensity score model posterior computation. The errors are plotted as a function of the number of CG iterations (bottom axis) and of the computational time relative to the direct linear algebra (top axis). The solid line shows the root mean squared residual $p^{-1/2}\|\tilde{\boldsymbol{r}}_k\|_2$ used in the stopping criteria (4.20). The dotted vertical line indicates when $p^{-1/2}\|\tilde{\boldsymbol{r}}_k\|_2$ reaches the threshold 10^{-6}. The 2nd-moment normalized errors (5.26) are shown as the blue dashed line for the prior preconditioner and orange dash-dot line for the Jacobi preconditioner.

5.6. Accuracy of CG sampler

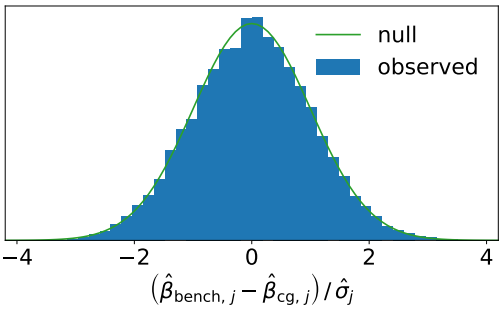
Since CG technically does not yield an exact solution when terminated after $k \ll p$ iterations, in this section we assess the accuracy of the samples generated by the CG-accelerated Gibbs. We compare these samples against those generated by the direct Gibbs, which we take as "ground truth." As we show, perturbation of the target distribution introduced by the CG approximation error is, if any, so small that it is essentially negligible.

In comparing the two sets of samples, we encountered one complication. While the global shrinkage parameter τ generally demonstrates good mixing under the Bayesian bridge prior (see Section 3.1), for our real world examples it proved difficult to obtain a sufficiently large effective sample size for τ within a reasonable amount of time. In order to ensure that the effective sample sizes for $\boldsymbol{\beta}$ are large enough to adequately characterize the stationary distribution, therefore, we employ an empirical Bayes approach. We first find a value $\hat{\tau}$ which approximately maximizes the marginal likelihood through Monte Carlo expectation-maximization (MCEM) algorithm (Casella, 2001). We then run the two samplers conditional on this value $\hat{\tau}$. The MCEM algorithm used here is essentially the same as in Park and Casella (2008) with only one difference — after sampling $\boldsymbol{\beta}^{(m)}$ and $\boldsymbol{\omega}^{(m)}$ from their conditionals, we update τ by maximizing the log density of $\tau \,|\, \boldsymbol{\beta}^{(m)}, \boldsymbol{\omega}^{(m)}, \boldsymbol{y}, \boldsymbol{X}$ with the local shrinkage parameter $\boldsymbol{\lambda}$ marginalized out.

We check for significant differences between the two sets of samples as follows. We first set $\hat{\boldsymbol{\beta}}_{\text{bench}}$ and $\hat{\boldsymbol{\beta}}_{\text{cg}}$ to be the posterior means estimated by averaging the samples from the direct Gibbs (used as a benchmark) and CG-accelerated Gibbs. Figure 5.6(a) graphically compares these two estimators as an informal sanity check. We then estimate the effective sample sizes of β_j from the respective samplers using the R CODA package (Plummer et al., 2006). These estimated effective sample sizes



(a) Comparison of the regression coefficient estimates (posterior means) between those based on the direct and CG-accelerated Gibbs samplers.



(b) Normalized histogram for the standardized differences $(\hat{\beta}_{\text{bench},j} - \hat{\beta}_{\text{cg},j})/\hat{\sigma}_i$, where $\hat{\sigma}_i^2$ is an estimate of the Monte Carlo variance of $\hat{\beta}_{\text{bench},j} - \hat{\beta}_{\text{cg},j}$. Normality of the histogram indicates no statistically significant difference between the two MCMC outputs.

Fig. 5.6. Diagnostic plots to check for any statistically significant differences in the two MCMC outputs.

can be used to estimate the Monte Carlo standard deviations $\hat{\sigma}_j$ of the differences $\hat{\beta}_{\text{bench},j} - \hat{\beta}_{\text{cg},j}$ (Geyer, 2011). When the two sets of MCMC samples have the same stationary distribution, the standardized differences $(\hat{\beta}_{\text{bench},j} - \hat{\beta}_{\text{cg},j})/\hat{\sigma}_j$ are approximately distributed as the standard Gaussians by the Markov chain central limit theorem (Geyer, 2011).

Figure 5.6(b) confirms that the distribution of the standardized differences closely follows the "null" distribution. Figure 5.6(a) and 5.6(b) are based on the posterior samples for the propensity score model (5.24), but we obtained essentially the same result under the treatment effect model (5.25). We additionally perform the same diagnostic on the estimators of the posterior second moment of $\boldsymbol{\beta}$ and obtained similar results.

5.7. Results and clinical conclusions from sparse regression models

The propensity score model finds that the patients treated by dabigatran and warfarin have substantial differences in their predictor characteristics. More precisely, a significant fraction of the individuals are much more likely to have been treated by one drug over the other as can be seen from Figure 5.7.† With the Bayesian approach, it is also straightforward to obtain uncertainty in the propensity score estimates. For instance, we can easily characterize uncertainty in the *inverse probability weight*, defined as a function of the propensity score p_i as $w_i = p_i^{-1}$ if in

†Figure 5.7 shows distributions of preference score, a transformation of propensity score that has been suggested as a more interpretable measure of the difference in covariate characteristics between the treated and control groups (Walker et al., 2013).

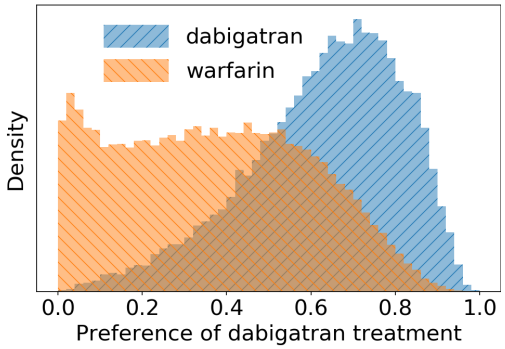


Fig. 5.7. Histogram (normalized to represent density) of the posterior means of preference scores for each of the two groups.

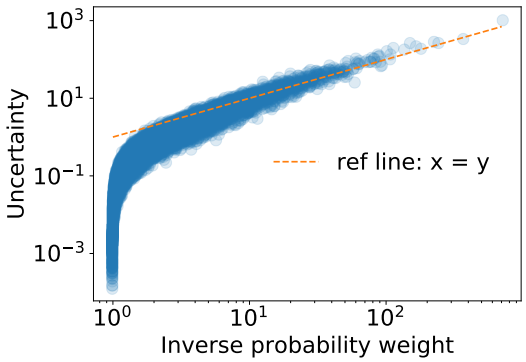


Fig. 5.8. Posterior means of the inverse probability weights, plotted against twice the posterior standard deviations as a measure of uncertainty. The plots are in the log-log scale and the dashed line indicates the coordinates $x = y$.

the treated group and $w_i = (1 - p_i)^{-1}$ if in the control group. Inverse probability weights are widely used due to their highly desirable theoretical properties, but have known issues of being unstable (Stuart, 2010). In fact, Figure 5.8 shows that the posterior uncertainties of the large inverse probability weights are as large as their posterior means.

Two of the most significant predictors are the year of the treatment and age group. Both predictors have been encoded as binary indicators in the design matrix for simplicity, but the categorical and ordinal predictors in our model could have been estimated with shrinkage priors analogous to (Bayesian) grouped or fused lasso (Hastie et al., 2009; Kyung et al., 2010; Xu et al., 2015). The posterior mean and 95% credible intervals of these regression coefficients are shown in Figure 5.9. Figure 5.9 shows the effect sizes relative to the year 2010 and the age group $65 \sim 69$; when actually fitting the model, however, we use the most common category as a baseline for categorical variables.

For the treatment effect model, Figure 5.10(a) shows the posterior distribution of the average treatment effect of dabigatran over warfarin. The posterior indicates an evidence, though not a conclusive one, for the lower incidence rate of intracranial hemorrhage attributable to dabigatran. This is consistent with findings of Graham et al. (2015). The violin plot of Figure 5.10(b) summarizes the posterior distributions of the baseline incident rates within each propensity score strata. While the differences in the baseline incident rates may seem insignificant because of the overlaps in their posterior marginals, there is significant covariation among the baseline incident rates. In fact, the baseline for the 2nd quintle has more than 95% posterior probability of being larger than those for the 1st and 5th quintile.

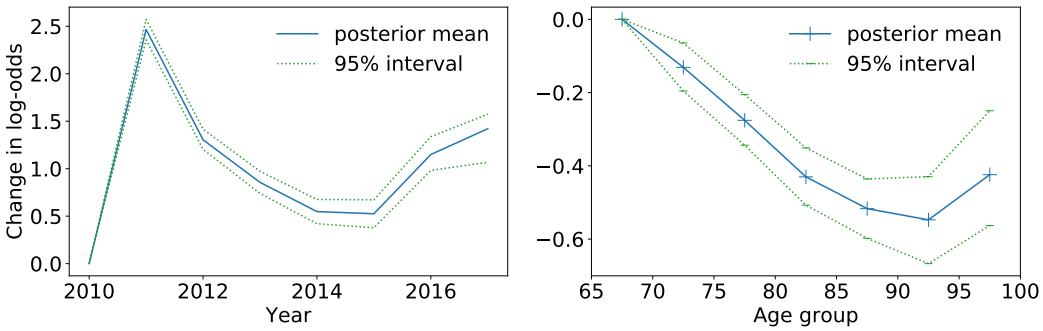
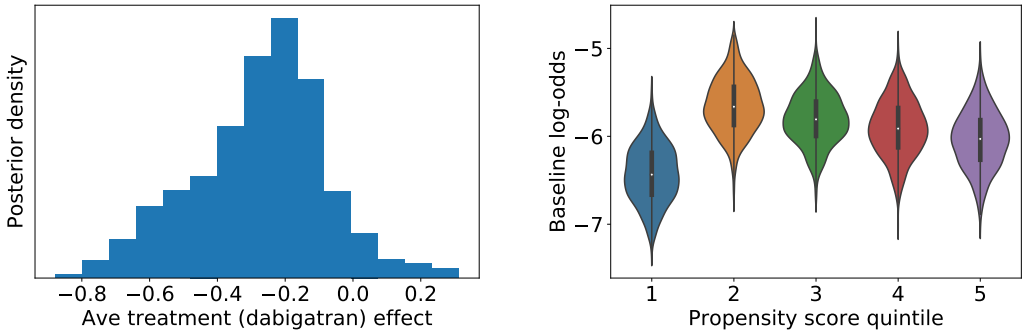


Fig. 5.9. Posterior means and 95% credible intervals for the regression coefficients of the treatment year and age group indicators. The age groups are divided into 5-year windows.



(a) Effect of treatment by dabigatran over warfarin on the incident rates of intracranial hemorrhage. Averaged over the study population.

(b) Violin plot of the baseline incident rates of intracranial hemorrhage within each propensity score strata.

Fig. 5.10. Posterior distributions from the treatment effect estimation model.

6. Acknowledgment

We would like to thank Yuxi Tian and Martijn Schuemie for their help in wrangling the data set used in Section 5. We also thank Jianfeng Lu and Ilse Ipsen for useful discussions on linear algebra topics. Finally, this work is partially supported by National Science Foundation grant DMS 1264153 and National Institutes of Health grant U19 AI135995.

Appendix A Proofs

We first derive Theorem 2.6 as a consequence of Theorem 2.5 before we proceed to prove Theorem 2.5.

Proof (Theorem 2.6). By Theorem B.4 below, the $(m + m')$-th CG iterate

$\boldsymbol{\beta}_{m+m'}$ satisfies the following bound:

$$\frac{\|\boldsymbol{\beta}_{m+m'} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq 2 \left(\frac{\sqrt{\nu_{m+1}/\nu_p} - 1}{\sqrt{\nu_{m+1}/\nu_p} + 1} \right)^{m'}, \tag{A.27}$$

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

$$1 \leq \nu_p \leq \nu_{m+1} \leq 1 + \min_{k+\ell=m} \tau^2 \lambda_{(k+1)}^2 \nu_{\ell+1}\big((\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X})_{(-k)}\big) = \widetilde{\kappa}_m \tag{A.28}$$

and hence that $\nu_{m+1}/\nu_p \leq \widetilde{\kappa}_m$. 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.27) in terms of $\widetilde{\kappa}_m$, yielding the desired inequality (2.15). □

Proof (Theorem 2.5). We prove the more general inequality (2.14). The lower bound $1 \leq \nu_{k+\ell}(\tilde{\boldsymbol{\Phi}})$ is an immediate consequence of Proposition A.1. For the upper bound, first note that $\nu_{k+\ell}(\tilde{\boldsymbol{\Phi}}) \leq \nu_\ell\Big(\tilde{\boldsymbol{\Phi}}_{(-k)}\Big)$ by the Poincaré separation theorem (Theorem A.2). From the expression (2.11) for $\tilde{\boldsymbol{\Phi}}$, we have

$$\begin{aligned} \nu_\ell\Big(\tilde{\boldsymbol{\Phi}}_{(-k)}\Big) &= \nu_\ell\Big(\boldsymbol{I}_k + \tau^2 \boldsymbol{\Lambda}_{(-k)} (\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X})_{(-k)} \boldsymbol{\Lambda}_{(-k)}\Big) \\ &= 1 + \tau^2\, \nu_\ell\Big(\boldsymbol{\Lambda}_{(-k)} (\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X})_{(-k)} \boldsymbol{\Lambda}_{(-k)}\Big), \end{aligned} \tag{A.29}$$

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

$$\nu_\ell\Big(\tilde{\boldsymbol{\Phi}}_{(-k)}\Big) \leq 1 + \tau^2 \lambda_{(k+1)}^2\, \nu_\ell\big((\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X})_{(-k)}\big). \tag{A.30}$$

Thus we have shown

$$\nu_{k+\ell}(\tilde{\boldsymbol{\Phi}}) \leq 1 + \tau^2 \lambda_{(k+1)}^2\, \nu_\ell\big((\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X})_{(-k)}\big) \leq 1 + \tau^2 \lambda_{(k+1)}^2\, \nu_\ell\big(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\big), \tag{A.31}$$

where the inequality $\nu_\ell\big((\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X})_{(-k)}\big) \leq \nu_\ell\big(\boldsymbol{X}^\mathsf{T}\boldsymbol{\Omega}\boldsymbol{X}\big)$ 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 symmetric matrix $\boldsymbol{A}$, let $\boldsymbol{A}_{(-k)}$ denote a sub-matrix with the first k rows and columns removed from $\boldsymbol{A}$. Then the eigenvalues of a symmetric matrix $\boldsymbol{A}$ and its sub-matrix $\boldsymbol{A}_{(-k)}$ satisfies*

$$\nu_{k+\ell}(\boldsymbol{A}) \leq \nu_\ell(\boldsymbol{A}_{(-k)}) \leq \nu_\ell(\boldsymbol{A}) \tag{A.32}$$

for any $\ell \geq 1$. Since permuting the rows and columns of $\boldsymbol{A}$ does not change its eigenvalues, the above inequality in fact holds for any sub-matrix of $\boldsymbol{A}$ obtained by removing k rows and columns of $\boldsymbol{A}$ corresponding to an arbitrary set of common row and column 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*

$$\nu_k(\boldsymbol{B}^{1/2}\boldsymbol{A}\boldsymbol{B}^{1/2}) \leq \nu_k(\boldsymbol{A}) \quad \textit{for} \quad k = 1, \ldots, p \tag{A.33}$$

where $\nu_k(\cdot)$ denotes the k-th 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 behind convergence behavior of CG

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.‡ 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 Section 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.

Section B.1 explains the critical first step in understanding the convergence of CG applied to a positive definite system $\boldsymbol{\Phi}\boldsymbol{\beta} = \boldsymbol{b}$ — relating the CG approximation error to a polynomial interpolation error over the set $\{\nu_1, \ldots, \nu_p\}$ consisting of the eigenvalues of $\boldsymbol{\Phi}$. Equipped with this perspective, one can understand Theorem 2.2 as a generic and rather crude bound, ignoring the distributions of ν_j'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 of $\boldsymbol{\Phi}$ are approximated by 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.

Section B.3 provides the proofs of all the results stated in this section.

‡For example, the discussions beyond Theorem 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).

B.1 CG approximation error as polynomial approximation error

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

$$\mathcal{P}_k = \{Q_k(\nu) : Q_k \text{ is a polynomial of degree } k \text{ with } Q_k(0) = 1\}. \tag{B.34}$$

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

PROPOSITION B.1. *The difference between the k-th CG iterate $\boldsymbol{\beta}_k$ and the exact solution $\boldsymbol{\beta}$ can be expressed as*

$$\boldsymbol{\beta}_k - \boldsymbol{\beta} = R_k(\boldsymbol{\Phi})\,(\boldsymbol{\beta}_0 - \boldsymbol{\beta}) \quad \textit{for } R_k = \operatorname*{argmin}_{Q_k \in \mathcal{P}_k} \|Q_k(\boldsymbol{\Phi})(\boldsymbol{\beta}_0 - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}. \tag{B.35}$$

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

$$\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}} \leq \|Q_k(\boldsymbol{\Phi})(\boldsymbol{\beta}_0 - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}. \tag{B.36}$$

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 k-th degree polynomial Q_k with the constraint $Q_k(0) = 1$.

THEOREM B.2.

$$\frac{\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq \min_{Q_k \in \mathcal{P}_k} \max_{j=1,\ldots,p} |Q_k(\nu_j)|, \tag{B.37}$$

where ν_j denotes the j-th 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 Characterizing CG approximation error via polynomial approximation

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

THEOREM B.3.

$$\min_{Q_k \in \mathcal{P}_k} \max_{\nu \in [\nu_{min}, \nu_{max}]} |Q_k(\nu)| \leq 2 \left(\frac{\sqrt{\nu_{max}/\nu_{min}} - 1}{\sqrt{\nu_{max}/\nu_{min}} + 1} \right)^k. \tag{B.38}$$

By setting $\nu_{\min} = \nu_p$ and $\nu_{\max} = \nu_1$, Theorem 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 $\{\nu_p, \ldots, \nu_1\}$, rather than the entire interval $[\nu_p, \nu_1]$, 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 $r < p$:*

$$\frac{\|\boldsymbol{\beta}_{r+k} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} = \min_{Q_{r+k} \in \mathcal{P}_{r+k}} \max_{j=1,\ldots,p} |Q_{r+k}(\nu_j)| \leq 2 \left(\frac{\sqrt{\nu_{r+1}/\nu_p} - 1}{\sqrt{\nu_{r+1}/\nu_p} + 1} \right)^k, \tag{B.39}$$

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 $\mathcal{P}_k$. Intuitively, this constraint makes the smallest eigenvalues more significant contributers to the polynomial interpolation error because it competes with the objective $Q_k(\nu) \approx 0$ near the smallest eigenvalues. 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.

In the context of CG, the Ritz values at the k-th CG iteration refer to the roots $\{\widehat{\nu}_1^{(k)}, \ldots, \widehat{\nu}_k^{(k)}\}$ of the optimal CG polynomial R_k as defined in (B.35). 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 to the largest and smallest eigenvalues of $\boldsymbol{\Phi}$ (Trefethen and Bau, 1997; Driscoll et al., 1998; Kuijlaars, 2006). More precisely, we have $\widehat{\nu}_i^{(k)} \to \nu_i$ for $i = 1, \ldots, r$ and $\widehat{\nu}_{k-i}^{(k)} \to \nu_{p-i}$ for $i = 0, \ldots, s$ as $k \to p$. While the convergence rates of the Ritz values can be shown to be exponential, unless $\max\{r, s\} \ll p$, in practice quite a large number of CG iterations may be required to obtain good approximations of the eigenvalues (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 k-th iterate satisfies the following bound:*

$$\frac{\|\boldsymbol{\beta}_{k+\ell} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \leq C_{k,r,s}\, 2 \left(\frac{\sqrt{\nu_{r+1}/\nu_{p-s}} - 1}{\sqrt{\nu_{r+1}/\nu_{p-s}} + 1} \right)^\ell, \tag{B.40}$$

where $C_{k,r,s} = C_{k,r,s}\big(\widehat{\nu}_1^{(k)}, \ldots, \widehat{\nu}_k^{(k)}\big) \to 1$ 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 $\boldsymbol{\Phi}$.

B.3 Proofs for Section B

PROOF (PROPOSITION B.1). As discussed in Section 2.2, the k-th CG iterate belongs to an affine space $\boldsymbol{\beta}_0 + \mathcal{K}(\boldsymbol{\Phi}, \boldsymbol{r}_0, k)$ with $\boldsymbol{r}_0 = \boldsymbol{\Phi}\boldsymbol{\beta}_0 - \boldsymbol{b} = \boldsymbol{\Phi}(\boldsymbol{\beta}_0 - \boldsymbol{\beta})$. An element $\boldsymbol{\beta}'$ of the affine space can be written as

$$\boldsymbol{\beta}' = \boldsymbol{\beta}_0 + \sum_{\ell=1}^{k} c_\ell \boldsymbol{\Phi}^{\ell-1} \boldsymbol{r}_0 = \boldsymbol{\beta} + (\boldsymbol{\beta}_0 - \boldsymbol{\beta}) + \sum_{\ell=1}^{k} c_\ell \boldsymbol{\Phi}^\ell (\boldsymbol{\beta}_0 - \boldsymbol{\beta}) \tag{B.41}$$

for some $c_1, \ldots, c_k$. In other words, for any $\boldsymbol{\beta}'$ in the affine space we can write

$$\boldsymbol{\beta}' - \boldsymbol{\beta} = Q_k(\boldsymbol{\Phi})\,(\boldsymbol{\beta}_0 - \boldsymbol{\beta}) \tag{B.42}$$

for some $Q_k \in \mathcal{P}_k$. Together with the optimality property (2.7) of the CG iterates, the representation (B.42) implies

$$\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}} = \|R_k(\boldsymbol{\Phi})(\boldsymbol{\beta}_0 - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}} = \min_{\boldsymbol{\beta}' \in \boldsymbol{\beta}_0 + \mathcal{K}(\boldsymbol{\Phi}, \boldsymbol{r}_0, k)} \|\boldsymbol{\beta}' - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}} \quad \square \tag{B.43}$$

Proof (Theorem B.2). Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p$ be the unit eigenvectors of $\boldsymbol{\Phi}$ associated with the eigenvalues $\nu_1, \ldots \nu_p$. By the spectral theorem for normal matrices (Section 2.5 of Horn and Johnson (2012)), the unit eigenvectors form an orthonormal basis. In particular, we can write $\boldsymbol{\beta}_0 - \boldsymbol{\beta} = \sum_{j=1}^p c_j \boldsymbol{v}_j$ for $c_j = \langle \boldsymbol{\beta}_0 - \boldsymbol{\beta}, \boldsymbol{v}_j \rangle$. Observe that, for any $Q_k \in \mathcal{P}_k$,

$$\boldsymbol{\Phi}^{1/2} Q_k(\boldsymbol{\Phi})\,(\boldsymbol{\beta}_0 - \boldsymbol{\beta}) = \textstyle\sum_{j=1}^p c_j \boldsymbol{\Phi}^{1/2} Q_k(\boldsymbol{\Phi}) \boldsymbol{v}_j = \sum_{j=1}^p c_j \nu_j^{1/2} Q_k(\nu_j) \boldsymbol{v}_j. \tag{B.44}$$

Together with (B.36), the above equality yields

$$\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}^2 \leq \sum_{j=1}^p c_j^2 \nu_j Q_k(\nu_j)^2 \leq \max_{j=1,\ldots,p} Q_k(\nu_j)^2 \left(\sum_{j=1}^p c_j^2 \nu_j \right). \tag{B.45}$$

The result (B.37) follows from the above inequality since $\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}^2 = \sum_{j=1}^p c_j^2 \nu_j$.

The sharpness of the upper bound is proven by explicitly constructing an initial vector that achieves the bound; see Greenbaum (1979). $\square$

Proof (Theorem B.3). We can construct a shifted and scaled Chebyshev polynomial $P_k \in \mathcal{P}_k$ such that $|P_k(\nu)|$ is bounded by the right-hand side of (B.38) over the interval $[\nu_{\min}, \nu_{\max}]$. See Saad (2011) for further details. $\square$

Proof (Theorem B.4). Let Q_k^* denote the minimizer of $\max_{j=r+1,\ldots,p} |Q_k(\nu_j)|$ over $\mathcal{P}_k$ and define

$$Q'_{r+k}(\nu) = Q_k^*(\nu) \prod_{i=1}^r \left(\frac{\nu_i - \nu}{\nu_i} \right). \tag{B.46}$$

Then Q'_{r+k} satisfies $Q'_{r+k}(\nu_j) = 0$ for $j = 1, \ldots, r$ and $Q'_{r+k}(\nu_j) \leq Q_k^*(\nu_j)$ for $j = r+1, \ldots, p$. In particular, Q'_{r+k} satisfies

$$\max_{j=1,\ldots,p} |Q'_{r+k}(\nu_j)| \leq \max_{j=r+1,\ldots,p} |Q_k^*(\nu_j)| = \min_{Q_k \in \mathcal{P}_k} \max_{j=r+1,\ldots,p} |Q_k(\nu_j)|, \tag{B.47}$$

where the equality holds as we chose Q_k^* to be the minimizer. From the above inequality, it follows that

$$\min_{Q_{r+k} \in \mathcal{P}_{r+k}} \max_{j=1,\ldots,p} |Q_{r+k}(\nu_j)| \leq \max_{j=1,\ldots,p} |Q'_{r+k}(\nu_j)| \leq \min_{Q_k \in \mathcal{P}_k} \max_{j=r+1,\ldots,p} |Q_k(\nu_j)|. \tag{B.48}$$

Since the maximum taken over an interval $[\nu_p, \nu_{r+1}]$ is larger than that over its subset, (B.48) immediately implies

$$\min_{Q_{r+k} \in \mathcal{P}_{r+k}} \max_{j=1,\ldots,p} |Q_{r+k}(\nu_j)| \leq \min_{Q_k \in \mathcal{P}_k} \max_{\nu \in [\nu_p, \nu_{r+1}]} |Q_k(\nu)|. \tag{B.49}$$

The desired inequality (B.39) now follows immediately by bounding the right-hand side of (B.49) via Theorem B.3. □

Proof (Theorem B.5). We first prove the bound (B.40) for $C_{k,r,s}$ as defined in (B.52) below. Let R_k be the optimal CG polynomial at the k-th iteration as defined in B.35. Since $R_k(0) = 1$, the polynomial $R_k(\nu)$ can be expressed in terms of its roots $\widehat{\nu}_1^{(k)}, \ldots, \widehat{\nu}_k^{(k)}$ as

$$R_k(\nu) = \prod_{i=1}^{k} \left(\frac{\widehat{\nu}_i^{(k)} - \nu}{\widehat{\nu}_i^{(k)}} \right). \tag{B.50}$$

Now consider $Q_k \in \mathcal{P}_k$ such that

$$Q_k(\nu) = \prod_{i=1}^{r} \left(\frac{\nu_i - \nu}{\nu_i} \right) \prod_{i=0}^{s-1} \left(\frac{\nu_{p-i} - \nu}{\nu_{p-i}} \right) \prod_{i=r+1}^{k-s} \left(\frac{\widehat{\nu}_i^{(k)} - \nu}{\widehat{\nu}_i^{(k)}} \right) \tag{B.51}$$

and define

$$C_{k,r,s} = \max_{j=r+1,\ldots,p-s} Q_k(\nu_j)/R_k(\nu_j). \tag{B.52}$$

As in the proof of Theorem B.2, write $\boldsymbol{\beta}_0 - \boldsymbol{\beta} = \sum_{j=1}^{p} c_j \boldsymbol{v}_j$ so that $\boldsymbol{\beta}_k - \boldsymbol{\beta} = \sum_{j=1}^{p} c_j R_k(\nu_j) \boldsymbol{v}_j$. Let $\boldsymbol{\beta}'_k$ be a modification of $\boldsymbol{\beta}_k$ such that

$$\boldsymbol{\beta}'_k - \boldsymbol{\beta} = \sum_{j=r+1}^{p-s} c_j R_k(\nu_j) \boldsymbol{v}_j. \tag{B.53}$$

Also, let $R'_\ell \in \mathcal{P}_\ell$ be a minimizer of $\|Q_\ell(\boldsymbol{\Phi})(\boldsymbol{\beta}'_k - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}$ over $Q_\ell \in \mathcal{P}_\ell$. The polynomial R'_ℓ can be interpreted as the optimal CG polynomial at the ℓ-th iteration as in (B.35) when the initial vector is taken to be $\boldsymbol{\beta}'_k$. By the property (B.36) of the CG iterates, we have

$$\|\boldsymbol{\beta}_{k+\ell} - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}} \leq \|R'_\ell(\boldsymbol{\Phi}) Q_k(\boldsymbol{\Phi})(\boldsymbol{\beta}_0 - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}. \tag{B.54}$$

We will now show that the right-hand side of (B.54) is bounded above by that of (B.40). By our definition of $\boldsymbol{\beta}'_k$ and $C_{k,r,s}$ in (B.53) and (B.52), we have

$$\begin{aligned} \|R'_\ell(\boldsymbol{\Phi}) Q_k(\boldsymbol{\Phi})(\boldsymbol{\beta}_0 - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}^2 &= \sum_{j=r+1}^{p-s} \nu_j c_j^2 R'_\ell(\nu_j)^2 Q_k(\nu_j)^2 \\ &\leq C_{k,r,s}^2 \sum_{j=r+1}^{p-s} \nu_j c_j^2 R'_\ell(\nu_j)^2 R_k(\nu_j)^2 \\ &= C_{k,r,s}^2 \|R'_\ell(\boldsymbol{\Phi})(\boldsymbol{\beta}'_k - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}^2. \end{aligned} \tag{B.55}$$

Noting that $\|\boldsymbol{\beta}'_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}} \le \|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}$, we obtain

$$\|R'_\ell(\boldsymbol{\Phi})Q_k(\boldsymbol{\Phi})(\boldsymbol{\beta}_0 - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}} \le C_{k,r,s}\frac{\|R'_\ell(\boldsymbol{\Phi})(\boldsymbol{\beta}'_k - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}'_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}. \tag{B.56}$$

Now, notice that $R'_\ell(\boldsymbol{\Phi})(\boldsymbol{\beta}'_k - \boldsymbol{\beta})$ is the residual of the ℓ-th CG iterate starting from the initial vector $\boldsymbol{\beta}'_k$. Therefore, by Lemma B.6 below combined with Theorem B.3, we have

$$\frac{\|R'_\ell(\boldsymbol{\Phi})(\boldsymbol{\beta}'_k - \boldsymbol{\beta})\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}'_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \le 2\left(\frac{\sqrt{\nu_{r+1}/\nu_{p-s}} - 1}{\sqrt{\nu_{r+1}/\nu_{p-s}} + 1}\right)^\ell. \tag{B.57}$$

The claimed inequality (B.40) now follows from (B.54), (B.56), and (B.57).

Now we turn to proving the claimed property of $C_{k,r,s}$. Note that

$$\frac{Q_k(\nu)}{R_k(\nu)} = \prod_{i=1}^{r}\frac{\widehat{\nu}_i^{(k)}}{\nu_i}\left(\frac{\nu_i - \nu}{\widehat{\nu}_i^{(k)} - \nu}\right)\prod_{i=0}^{s-1}\frac{\widehat{\nu}_{k-i}^{(k)}}{\nu_{p-i}}\left(\frac{\nu_{p-i} - \nu}{\widehat{\nu}_{k-i}^{(k)} - \nu}\right). \tag{B.58}$$

The rest of the proof focuses on the case $r = 1$ and $s = 0$ for clarity's sake; the proof remains essentially identical in the general case except for extra notational clutters. Under this case, we have

$$\max_{j=2,\ldots,p}\left|\frac{Q_k(\nu_j)}{R_k(\nu_j)}\right| = \frac{\widehat{\nu}_1^{(k)}}{\nu_1}\max_{j=2,\ldots,p}\left|\frac{\nu_1 - \nu_j}{\widehat{\nu}_1^{(k)} - \nu_j}\right| = \frac{\widehat{\nu}_1^{(k)}}{\nu_1}\max_{j=2,\ldots,p}\left|1 - \frac{\widehat{\nu}_1^{(k)} - \nu_1}{\nu_j - \nu_1}\right|^{-1}. \tag{B.59}$$

Provided $|\widehat{\nu}_1^{(k)} - \nu_1| = \min_{j=1,\ldots,p}|\widehat{\nu}_1^{(k)} - \nu_j|$, the above inequality simplifies to

$$\max_{j=2,\ldots,p}\left|\frac{Q_k(\nu_j)}{R_k(\nu_j)}\right| = \frac{\widehat{\nu}_1^{(k)}}{\nu_1}\left|1 - \frac{\widehat{\nu}_1^{(k)} - \nu_1}{\nu_2 - \nu_1}\right|^{-1}. \tag{B.60}$$

So we have $C_{k,r,s} \to 1$ as $\widehat{\nu}_1^{(k)} \to \nu_1$ in the case $r = 1$ and $s = 0$. □

LEMMA B.6. *Let $(\nu_j, \boldsymbol{v}_j)$ for $j = 1, \ldots, p$ denote the eigenvalue and eigenvector pairs of $\boldsymbol{\Phi}$. If the initial vector $\boldsymbol{\beta}_0$ satisfies $\langle \boldsymbol{\beta}_0 - \boldsymbol{\beta}, \boldsymbol{v}_j\rangle = 0$ for $j \in J \subset \{1, \ldots, p\}$, then the bound* (B.37) *holds over the set $\{1, \ldots, p\} \setminus J$ i.e.*

$$\frac{\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}}{\|\boldsymbol{\beta}_0 - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}} \le \min_{Q_k \in \mathcal{P}_k}\max_{j \notin J}|Q_k(\nu_j)|. \tag{B.61}$$

PROOF. The proof is identical to that of Theorem B.2 except that we can replace the bound (B.45) with

$$\|\boldsymbol{\beta}_k - \boldsymbol{\beta}\|_{\boldsymbol{\Phi}}^2 \le \sum_{j \notin J} c_j^2 \nu_j Q_k(\nu_j)^2 \le \max_{j \notin J} Q_k(\nu_j)^2\left(\sum_{j \notin J} c_j^2 \nu_j\right) \tag{B.62}$$

since $c_j = 0$ for $j \in J$ by assumption. □

References

Armagan, A., Dunson, D. B. and Lee, J. (2013) Generalized double Pareto shrinkage. *Statistica Sinica*, **23**, 119.

Berger, J. O., Bernardo, J. M., Sun, D. et al. (2009) The formal definition of reference priors. *The Annals of Statistics*, **37**, 905–938.

Bhadra, A., Datta, J., Polson, N. G., Willard, B. et al. (2017) The horseshoe+ estimator of ultra-sparse signals. *Bayesian Analysis*, **12**, 1105–1131.

Bhattacharya, A., Chakraborty, A. and Mallick, B. K. (2016) Fast sampling with Gaussian scale mixture priors in high-dimensional regression. *Biometrika*, **103**, 985–991.

Bhattacharya, A., Pati, D., Pillai, N. S. and Dunson, D. B. (2015) Dirichlet–Laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, **110**, 1479–1490.

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2010) The horseshoe estimator for sparse signals. *Biometrika*, **97**, 465–480.

Casella, G. (2001) Empirical Bayes Gibbs sampling. *Biostatistics*, **2**, 485–500.

Cockayne, J., Oates, C. and Girolami, M. (2018) A Bayesian conjugate gradient method. *arXiv:1801.05242*.

Datta, J., Ghosh, J. K. et al. (2013) Asymptotic properties of Bayes risk for the horseshoe prior. *Bayesian Analysis*, **8**, 111–132.

Dongarra, J., Heroux, M. A. 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.

Geyer, C. (2011) Introduction to Markov chain Monte Carlo. In *Handbook of Markov Chain Monte Carlo*, 3–48. CRC Press.

Ghosh, J., Li, Y., Mitra, R. et al. (2018) On the use of Cauchy prior distributions for Bayesian logistic regression. *Bayesian Analysis*, **13**, 359–383.

Gibbs, M. and MacKay, D. (1996) Effcient implementation of Gaussian processes. Unpublished manuscript.

Golub, G. H. and Van Loan, C. F. (2012) *Matrix computations*, vol. 3. Johns Hopkins University Press.

Graham, D. J., Reichman, M. E., Wernecke, M., Zhang, R., Southworth, M. R., Levenson, M., Sheu, T.-C., Mott, K., Goulding, M. R., Houstoun, M. et al. (2015) Cardiovascular, bleeding, and mortality risks in elderly Medicare patients treated with dabigatran or warfarin for non-valvular atrial fibrillation. *Circulation*, **131**, 157–164.

Hahn, P. R., He, J. and Lopes, H. F. (2018) Efficient sampling for Gaussian linear regression with arbitrary priors. *Journal of Computational and Graphical Statistics*.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning*. Springer Series in Statistics. Springer.

Hestenes, M. R. and Stiefel, E. (1952) Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, **49**.

Horn, R. A. and Johnson, C. R. (2012) *Matrix Analysis*. Cambridge University Press.

Hripcsak, G., Duke, J. D., Shah, N. H., Reich, C. G., Huser, V., Schuemie, M. J., Suchard, M. A., Park, R. W., Wong, I. C. K., Rijnbeek, P. R. et al. (2015) Observational Health Data Sciences and Informatics (OHDSI): Opportunities for observational researchers. *Studies in health technology and informatics*, **216**, 574.

Johndrow, J. E., Orenstein, P. and Bhattacharya, A. (2018) Bayes shrinkage at GWAS scale: Convergence and approximation theory of a scalable MCMC algorithm for the horseshoe prior. *arXiv:1705.00841*.

Jolliffe, I. T. (2002) *Principal Component Analysis*. Springer Series in Statistics. Springer.

Kyung, M., Gill, J., Ghosh, M., Casella, G. et al. (2010) Penalized regression, standard errors, and Bayesian lassos. *Bayesian Analysis*, **5**, 369–411.

Lanczos, C. (1952) Solution of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, **49**, 33–53.

Pal, S., Khare, K. et al. (2014) Geometric ergodicity for Bayesian shrinkage models. *Electronic Journal of Statistics*, **8**, 604–645.

Park, T. and Casella, G. (2008) The Bayesian Lasso. *Journal of the American Statistical Association*, **103**, 681–686.

Piironen, J. and Vehtari, A. (2017) Sparsity information and regularization in the horseshoe and other shrinkage priors. *arXiv:1707.01694*.

Plummer, M., Best, N., Cowles, K. and Vines, K. (2006) Coda: convergence diagnosis and output analysis for MCMC. *R news*, **6**, 7–11.

Polson, N. G. and Scott, J. G. (2010) Shrink globally, act locally: Sparse Bayesian regularization and prediction. *Bayesian Statistics*, **9**, 501–538.

Polson, N. G., Scott, J. G. and Windle, J. (2013) Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American Statistical Association*, **108**, 1339–1349.

— (2014) The Bayesian bridge. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **76**, 713–733.

Ripley, B. D. (1987) *Stochastic simulation*. New York, NY, USA: John Wiley & Sons.

Saad, Y. (2003) *Iterative Methods for Sparse Linear Systems*, vol. 82. Society for Industrial and Applied Mathematics.

Schuemie, M. J., Ryan, P. B., Hripcsak, G., Madigan, D. and Suchard, M. A. (2018a) 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.

— (2018b) A systematic approach to improving the reliability and scale of evidence from health care data. *arXiv:1803.10791*.

Stein, M. L., Chen, J. and Anitescu, M. (2012) Difference filter preconditioning for large covariance matrices. *SIAM Journal on Matrix Analysis and Applications*, **33**, 52–72.

Stroud, J. R., Stein, M. L. 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.

Stuart, E. A. (2010) Matching methods for causal inference: A review and a look forward. *Statistical Science*, **25**, 1.

Sun, Y. and Stein, M. L. (2016) Statistically and computationally efficient estimating equations for large spatial datasets. *Journal of Computational and Graphical Statistics*, **25**, 187–208.

Tian, Y., Schuemie, M. J. and Suchard, M. A. (2018) elll. *International Journal of Epidemiology*.

Trefethen, L. N. and Bau, D. (1997) *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics.

Walker, A. M., Patrick, A. R., Lauer, M. S., Hornbrook, M. C., Marin, M. G., Platt, R., Roger, V. L., Stang, P. and Schneeweiss, S. (2013) A tool for assessing the feasibility of comparative effectiveness research. *Comparative Effectiveness Research*, **3**, 11–20.

Xu, X., Ghosh, M. et al. (2015) Bayesian variable selection and estimation for group lasso. *Bayesian Analysis*, **10**, 909–936.

Zhang, L., Datta, A. and Banerjee, S. (2018) Practical Bayesian modeling and inference for massive spatial datasets on modest computing environments. *arXiv:1802.00495*.

Zhou, Q. and Guan, Y. (2017) Fast model-fitting of Bayesian variable selection regression using the iterative complex factorization algorithm. *arXiv preprint arXiv:1706.09888*.

Zucknick, M., Saadati, M. and Benner, A. (2015) Nonidentical twins: comparison of frequentist and Bayesian lasso for Cox models. *Biometrical Journal*, **57**, 959–981.

Supplement to "Prior-preconditioned conjugate gradient for accelerated Gibbs sampling in "large n & large p" sparse Bayesian logistic regression models"

S1. Further look at the CG convergence behavior in Section 3.3

The CG convergence behavior as illustrated in Figure 3.1 remains qualitatively consistent across different random draws of the right-hand vector $\boldsymbol{b}$ and across various metrics of the approximation error. Figure S1 shows the average of the coordinate-wise relative error as a function of the CG iterations as in Figure 3.1, but with an individual line for each of the random draws of $\boldsymbol{b}$. The convergence behaviors under the prior and Jacobi preconditioners are plotted in the two separate sub-figures to avoid cluttering the plot with too many lines. Figure S2 shows the CG convergence behaviors under the two additional error metrics: the ℓ^2-norm and $\boldsymbol{\Phi}$-norm distance between $\boldsymbol{\beta}_k$ and $\boldsymbol{\beta}_{\text{direct}}$.

S2. Additional discussion on using CG sampler for sparse regression

Throughout this section, we write $\boldsymbol{vw}$ and $\boldsymbol{v}/\boldsymbol{w}$ to denote an element-wise multiplication and division of two vectors $\boldsymbol{v}$ and $\boldsymbol{w}$.

S2.1. Additional details on the initial vector for CG iterations

As an alternative to the initialization $\boldsymbol{\beta}_0 = \boldsymbol{0}$, we consider three approaches for constructing the initial vector to the CG sampler. At the m-th Gibbs update, the CG

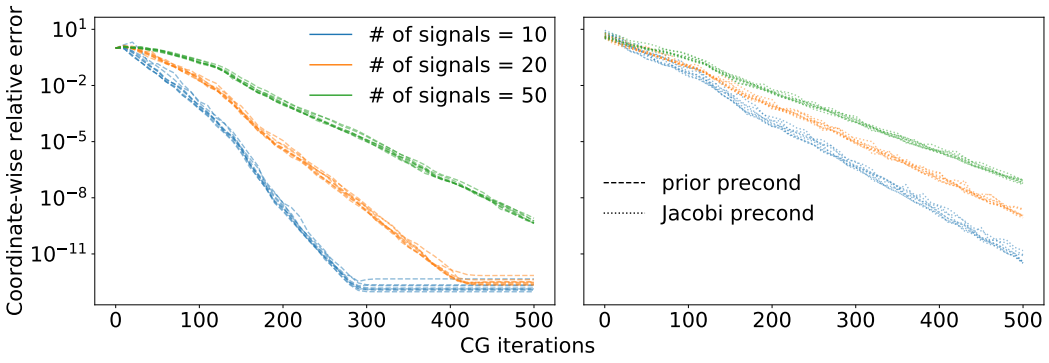


Fig. S1. Plots of the CG approximation errors (with the same error metric as used in Figure 3.1) as a function of the number of CG iterations. Shown on the left is under the prior preconditioner and on the right is under the Jacobi preconditioner. The three different colors corresponds to the three different posterior conditional distributions of $\boldsymbol{\beta}$ with the varying numbers of true signals. Within the same color, the different lines correspond to the different random draws of the right-hand vector $\boldsymbol{b}$ generated as in (2.4).

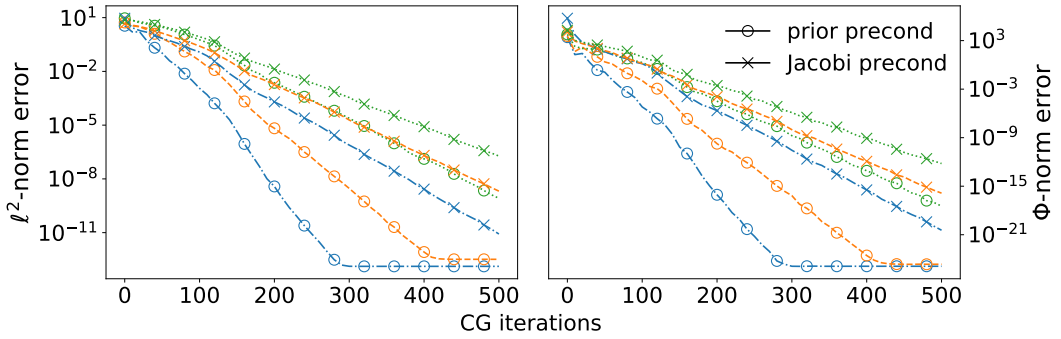


Fig. S2. Plots of the ℓ^2-norm (on the left) and the $\boldsymbol{\Phi}$-norm (on the right) between $\boldsymbol{\beta}_k$ and $\boldsymbol{\beta}_{\text{direct}}$ as a function of the number of CG iterations. Other than the use of the two alternative error metrics for the y-axes, each of the plotted lines directly corresponds to the one with the same color and marker in Figure 3.1.

sampler needs to draw $\boldsymbol{\beta}^{(m)}$ from the distribution $\boldsymbol{\beta} \,|\, \boldsymbol{\omega}^{(m-1)}, \boldsymbol{\lambda}^{(m-1)}, \tau^{(m-1)}, \boldsymbol{y}, \boldsymbol{X}$. As we have no control over the variability in $\boldsymbol{\beta}^{(m)}$, we focus on getting $\boldsymbol{\beta}_0$ as close as possible to the mean of $\boldsymbol{\beta}^{(m)}$. The two seemingly obvious choices of $\boldsymbol{\beta}_0$ are 1) the previous MCMC sample $\boldsymbol{\beta}^{(m-1)}$ and 2) the MCMC estimate $m^{-1}\sum_{i=0}^{m-1}\boldsymbol{\beta}^{(i)}$ of the expectation $\mathbb{E}[\boldsymbol{\beta} \,|\, \boldsymbol{y}, \boldsymbol{X}]$. These options, however, ignores the fact that the distribution of $\boldsymbol{\beta}^{(m)}$ depends strongly on $\tau^{(m-1)}\boldsymbol{\lambda}^{(m-1)}$, which generally is very different from $\tau^{(m-i)}\boldsymbol{\lambda}^{(m-i)}$ for $i \geq 2$.

We found the following approach, used in our CG-accelerated Gibbs samplers, to yield a better estimate of the mean and hence a better initialization for $\boldsymbol{\beta}^{(m)}$. We first estimate $\mathbb{E}[\tau^{-1}\boldsymbol{\lambda}^{-1}\boldsymbol{\beta} \,|\, \boldsymbol{y}, \boldsymbol{X}]$ by the estimator $\tilde{\boldsymbol{\beta}}_0 = m^{-1}\sum_{i=0}^{m-1}\boldsymbol{\beta}^{(i)}/\tau^{(i-1)}\boldsymbol{\lambda}^{(i-1)}$, where we define $\tau^{(-1)}\lambda_j^{(-1)} = 1$. (We sample $\boldsymbol{\beta}^{(i)}$ conditional on $\tau^{(i-1)}$ and $\boldsymbol{\lambda}^{(i-1)}$; see Section S3.1 for further details on our updating orders.) Then we rescale it with the current conditioned values of τ and $\boldsymbol{\lambda}$, setting $\boldsymbol{\beta}_0 = \tau^{(m-1)}\boldsymbol{\lambda}^{(m-1)}\tilde{\boldsymbol{\beta}}_0$ to obtain the initial vector. We compared this approach to the other two through a simulation study and found our choice to consistently yield smaller $\boldsymbol{\Phi}$-norm errors and faster convergence.

S2.2. Additional details on the termination criteria for CG

To relate the norm of the prior-preconditioned residual $\tilde{\boldsymbol{r}}_k = \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{b}}$ to the error metric (4.21), note that $\boldsymbol{\beta}_k - \boldsymbol{\beta} = \boldsymbol{M}^{-1/2}\tilde{\boldsymbol{\Phi}}^{-1}\tilde{\boldsymbol{r}}_k$ with $\boldsymbol{M} = \tau^{-2}\boldsymbol{\Lambda}^{-2}$. So we have

$$\left\|\boldsymbol{\xi}^{-1}\left(\boldsymbol{\beta}_k - \boldsymbol{\beta}\right)\right\|_2 = \left\|\boldsymbol{\xi}^{-1}(\tau\boldsymbol{\lambda})\left(\tilde{\boldsymbol{\Phi}}^{-1}\tilde{\boldsymbol{r}}_k\right)\right\|_2 \leq \left(\max_j \xi_j^{-1}\tau\lambda_j\right)\left\|\tilde{\boldsymbol{\Phi}}^{-1}\tilde{\boldsymbol{r}}_k\right\|_2, \tag{S1}$$

It is worth noting that the inequality in the above equation only represents the worst-case scenario; in more typical settings, one expects the norm of $\boldsymbol{\xi}^{-1}(\tau\boldsymbol{\lambda})\boldsymbol{v}$ to be related to that of $\boldsymbol{v}$ through some averages of $\xi_j^{-1}\tau\lambda_j$'s.

We now analyze a typical behavior of $\xi_j^{-1}\tau\lambda_j$ as the parameters $\boldsymbol{\omega}, \boldsymbol{\lambda}, \tau$ are drawn

from a sparse regression posterior. As before, we interpret $\tau^2\lambda_j^2$ as the prior variance of β_j (conditional on $\boldsymbol{\omega}, \boldsymbol{\lambda}, \tau$) before observing $\boldsymbol{y}, \boldsymbol{X}$. Note that

$$(\xi_j^{-1}\tau\lambda_j)^2 = \frac{\tau^2\lambda_j^2}{\mu_j^2 + \sigma_j^2}, \tag{S2}$$

where μ_j and σ_j are the conditional mean and variance of $\beta_j \,|\, \boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X}$. So the quantity $\xi_j^{-1}\tau\lambda_j$ is not too far from 1 if either $|\mu_j|$ or σ_j is in the same order of magnitude as $\tau\lambda_j$. If β_j's posterior is dominated by the prior shrinkage, we can expect the posterior (conditional) variance to be not much smaller than the prior one and hence $\sigma_j \approx \tau\lambda_j$. Otherwise, if $\sigma_j \ll \tau\lambda_j$ and the likelihood is a dominant contributer to the posterior, then the posterior of $\tau\lambda_j$ should concentrate around typical values of $|\mu_j|$ to maximize the marginal likelihood of $\beta_j \approx \mu_j$. Either way, we can expect $\xi_j^{-1}\tau\lambda_j$ to be in the same order of magnitude as 1.

From the relation (S1) and our analysis above, we deduce that

$$\left\|\boldsymbol{\xi}^{-1}\left(\boldsymbol{\beta}_k - \boldsymbol{\beta}\right)\right\|_2 \lessapprox \left\|\tilde{\boldsymbol{\Phi}}^{-1}\tilde{\boldsymbol{r}}_k\right\|_2 \leq \|\tilde{\boldsymbol{r}}_k\|_2, \tag{S3}$$

where the latter inequality follows from the fact that the largest eigenvalue of the prior-preconditioned matrix $\tilde{\boldsymbol{\Phi}}^{-1}$ is bounded above by 1 by Theorem 2.5.

S2.3. Details on preconditioning under uninformative priors

Consider a preconditioned matrix $\tilde{\boldsymbol{\Phi}} = \boldsymbol{M}^{-1/2}\boldsymbol{\Phi}\boldsymbol{M}^{-1/2}$ with $\boldsymbol{\Phi}$ as in (4.22) and $\boldsymbol{M} = \mathrm{diag}(\boldsymbol{\gamma}^{-2}, \tau^{-2}\boldsymbol{\lambda}^{-2})$. Let $\tilde{\boldsymbol{\Phi}}_{(-q-1)}$ denote the sub-matrix with the first $q+1$ rows and columns of $\tilde{\boldsymbol{\Phi}}$ removed. As shown in Section 2.4, the sub-matrix $\tilde{\boldsymbol{\Phi}}_{(-q-1)}$ has an eigenvalue distribution particularly well-suited to induce rapid CG convergence. By the Poincaré separation theorem (Theorem A.2), all but $q+1$ eigenvalues of the original matrix $\tilde{\boldsymbol{\Phi}}$ lie within the largest and smallest eigenvalues of the sub-matrix $\tilde{\boldsymbol{\Phi}}_{(-q-1)}$. In choosing γ_j's, we are therefore concerned with the behavior of the $q+1$ additional eigenvalues introduced by the unshrunk coefficients. Additionally, we should err on the side of introducing larger eigenvalues than smaller ones as the small eigenvalues impact CG convergence rates more significantly (Rule of Thumb 2.4).

With the above objectives in mind, we propose a choice

$$\gamma_j = c\,\widehat{\psi}_j \quad \text{for} \quad \widehat{\psi}_j^2 \approx \mathrm{var}(\beta_j \,|\, \boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X}) \tag{S4}$$

with $c \geq 1$, which is slightly different from the choice (4.23) presented in Section 4.3. We explain first the reasoning behind the choice (S4) and then why we can use (4.23) instead.

Let $\boldsymbol{\beta}_q = (\beta_0, \ldots, \beta_q)$ and $\boldsymbol{\beta}_{(-q)} = (\beta_{q+1}, \ldots, \beta_p)$. The smallest eigenvalues of $\tilde{\boldsymbol{\Phi}}$ correspond to the largest variances (conditional on $\boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X}$) of the Gaussian vector $\boldsymbol{M}^{1/2}\boldsymbol{\beta} = (\boldsymbol{\gamma}^{-1}\boldsymbol{\beta}_q, \tau^{-1}\boldsymbol{\lambda}^{-1}\boldsymbol{\beta}_{(-q)})$ along its principal components. We know that the variances of $\tau^{-1}\boldsymbol{\lambda}^{-1}\boldsymbol{\beta}_{(-q)}$ conditional on $\boldsymbol{\gamma}^{-1}\boldsymbol{\beta}_q$ are bounded above by 1 along any directions because the eigenvalues of the conditional precision matrix $\tilde{\boldsymbol{\Phi}}_{(-q)}$ are bounded below by 1. Therefore, we do not expect $(\boldsymbol{\gamma}^{-1}\boldsymbol{\beta}_q, \tau^{-1}\boldsymbol{\lambda}^{-1}\boldsymbol{\beta}_{(-q)})$

to have variances much larger than 1 unless the marginal variances of $\boldsymbol{\gamma}^{-1}\boldsymbol{\beta}_q$ are large. The proposed choice of γ_j's ensure that the marginal variances of $\gamma_j^{-1}\beta_j$'s are less than c^{-1}, up to the error in estimating $\mathrm{var}(\beta_j \,|\, \boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X})$ by $\widehat{\psi}_j^2$, and thus prevent an introduction of small eigenvalues to $\tilde{\boldsymbol{\Phi}}$. The multiplicative factor $c \geq 1$ provide an additional safeguard as we are less concerned about introducing large eigenvalues to $\boldsymbol{\Phi}$.

As the parameters $\boldsymbol{\omega}, \tau, \lambda$ are constantly updated during Gibbs sampling, technically we cannot estimate $\mathrm{var}(\beta_j \,|\, \boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X})$ from earlier MCMC samples. This is why in practice we use the choice (4.23) based on an estimator $\widehat{\eta}_j^2$ of $\mathrm{var}(\beta_j \,|\, \boldsymbol{y}, \boldsymbol{X})$. Using (4.23) in place of (S4) is justified in two ways. First, by the variance decomposition formula we have

$$\mathbb{E}_{\boldsymbol{\omega}, \tau, \boldsymbol{\lambda} \,|\, \boldsymbol{y}, \boldsymbol{X}} \left[\mathrm{var} \left(\beta_j \,|\, \boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X} \right) \right] \leq \mathrm{var}(\beta_j \,|\, \boldsymbol{y}, \boldsymbol{X}). \tag{S5}$$

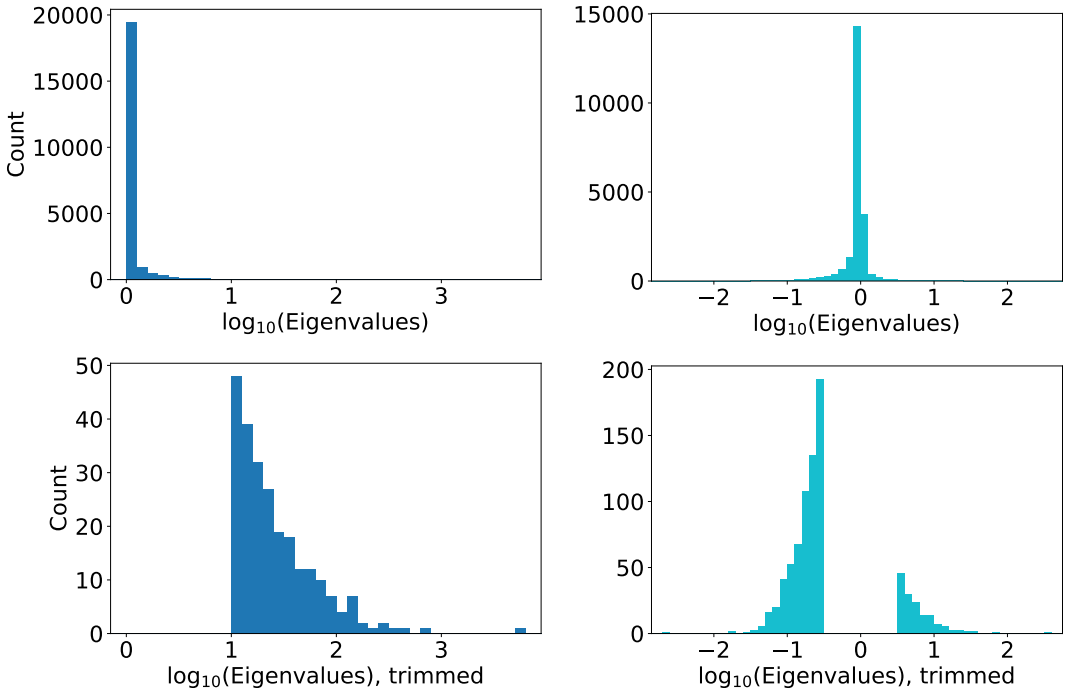
In other words, on average $\widehat{\eta}_j$ is an overestimate of $\widehat{\psi}_j$ which, as we have discussed, is more preferable to an underestimate. Secondly, the unshrunk coefficients $\beta_0, \ldots, \beta_q$ have only limited dependency on the shrinkage parameters τ and $\boldsymbol{\lambda}$ through $\beta_{q+1}, \ldots, \beta_p$. Also, in our experience we have never noticed any obvious correlations between the posterior samples of $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$. For these reasons, we suspect that $\mathrm{var}(\beta_j \,|\, \boldsymbol{y}, \boldsymbol{X})$ is generally not too far from $\mathrm{var}(\beta_j \,|\, \boldsymbol{\omega}, \boldsymbol{\lambda}, \tau, \boldsymbol{y}, \boldsymbol{X})$.

As mentioned in Section 4.3, once chosen within reasonable ranges, the precise values of γ_j's have rather small effect on the CG convergence rate. In our simulations (not presented in the manuscript), we found the delay in the CG convergence to be no more than $20 \sim 30\%$ even when the values of γ_j's were off by two orders of magnitude from empirically-determined optimal values. The convergence rate achieved by the proposed choice of $\boldsymbol{\gamma}$ was essentially indistinguishable from that achieved by an optimal choice.

S3. More detailed look at the CG-acceleration mechanism in the examples of Section 5

We saw in Figure 5.5 that the advantage of the prior preconditioner over the Jacobi one continues to hold for the propensity score model example. The difference in the CG convergence behaviors can again be explained by the eigenvalue distributions of the respective preconditioned matrices as shown in Figure S3. As in the simulated data examples of Section 3, the prior preconditioning leads to a tighter cluster of the eigenvalues and avoids introducing small eigenvalues. These features of the eigenvalue distribution translates into faster CG convergence (Rule of Thumb 2.4).

Figure S4 shows the number of the CG iterations, or equivalently of the matrix vector operations $\boldsymbol{v} \to \boldsymbol{\Phi}\boldsymbol{v}$, required to meet the convergence criteria (4.20) during each conditional update of $\boldsymbol{\beta}$ within the Gibbs sampler. First, the values of the global shrinkage parameter τ shown in the red dotted line deserves some explanations. As mentioned earlier, τ is updated via the Monte Carlo expectation-maximization step and eventually converges to a value approximately maximizing



(a) Under the prior preconditioner.

(b) Under the Jacobi preconditioner.

Fig. S3. Histograms of the eigenvalues of the preconditioned matrices as in Figure 3.2. The only differences here are that 1) the preconditioned matrices are based on a posterior sample from the propensity score model (5.24) and 2) the trimmed version for the Jacobi preconditioner removes the eigenvalues in the range $[-0.5, 0.5]$ as this choice better demonstrates the tail behavior here.

the marginal likelihood. With our initialization of the Gibbs sampler (see Section S3.1 for further details), τ starts out small before eventually converging to a larger value. Correspondingly, $\boldsymbol{\beta}$ starts out with most of its component shrunken to 0 before some of them eventually escape the shrinkage and converge to values away from 0.

As the CG sampler converges quicker when the conditional distribution of $\boldsymbol{\beta}$ is concentrated on sparser vectors (Theorem 2.6), the number of CG iterations until convergence is highly correlated with the value of τ in Figure S4. Other than the effects of fluctuation in τ, however, we can see that the conditional updates of $\boldsymbol{\beta}$ requires similar numbers of CG iterations despite the randomness in $\boldsymbol{\lambda}$, and $\boldsymbol{\omega}$.

While we have so far used the propensity score model example in analyzing the mechanism behind the CG-acceleration, essentially identical conclusions follow from the treatment effect model example as well. The prior preconditioning is again superior to the Jacobi one. The number of required CG iterations fluctuates along with the sparsity structure of $\boldsymbol{\beta} \,|\, \boldsymbol{\omega}, \tau, \boldsymbol{\lambda}, \boldsymbol{y}, \boldsymbol{X}$ but not significantly so after

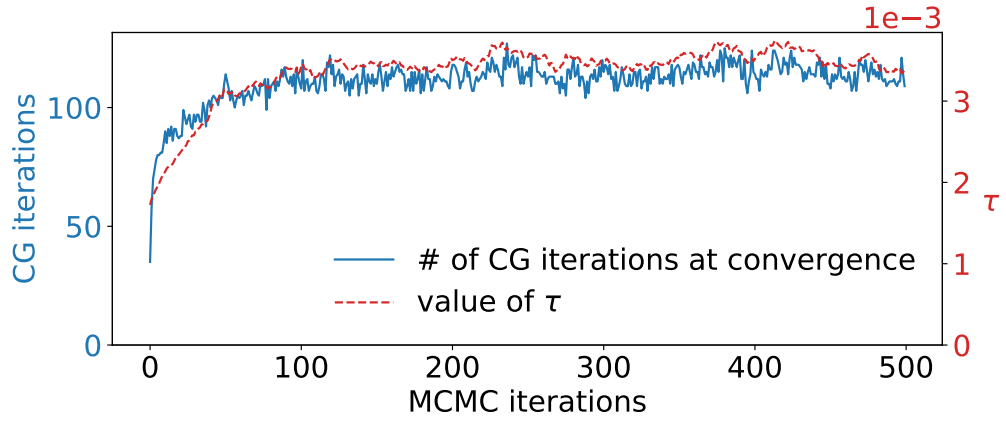


Fig. S4. Plot of the number of CG iterations required to meet the convergence criteria (4.20) during each update of $\boldsymbol{\beta}$ by the CC-accelerated Gibbs sampler. The target distribution here comes from the propensity score model (5.24). Also shown is the value of the global shrinkage parameter τ updated via the Monte Carlo expectation-maximization steps. The value of τ significantly affect the sparsity in the conditional distribution of $\boldsymbol{\beta}$ and hence the number of required CG iterations.

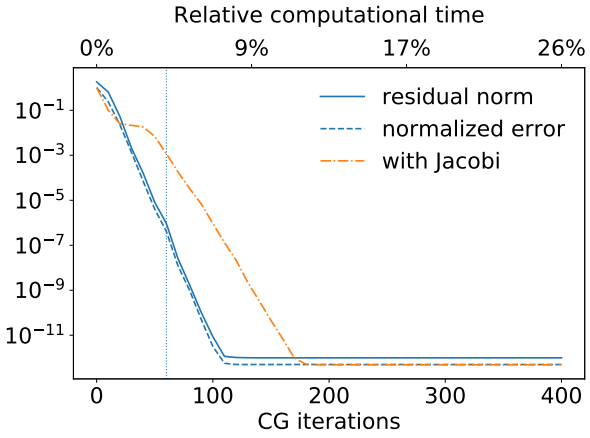
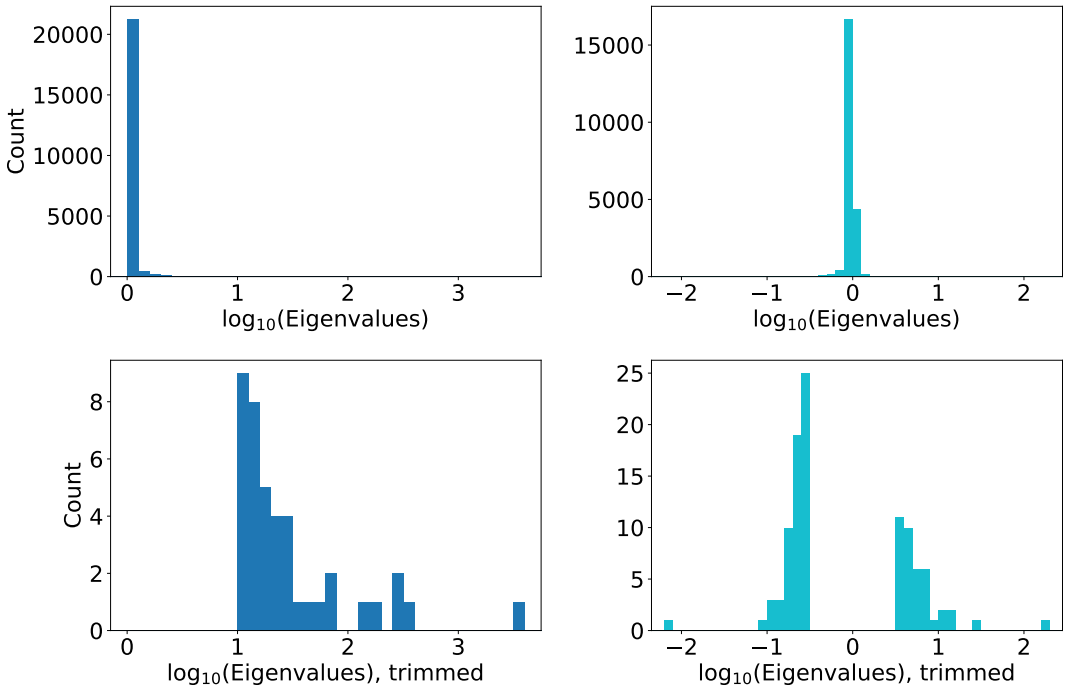


Fig. S5. Plot as in Figure 5.5 of the CG errors during a conditional update of $\boldsymbol{\beta}$. The only difference is that the plot here is based on the treatment effect model posterior computation.

the Markov chain converges — see Figure S5, S6, and S7.

S3.1. Further details on the Gibbs sampler initialization

We initialize the Gibbs sampler with the shrinkage parameter values of $\tau^{(0)} = 0.01$ and $\lambda_j^{(0)} = 1$. We then set $\boldsymbol{\omega}^{(0)}$ to the mean of its posterior distribution conditional on $\tau^{(0)}$, $\boldsymbol{\lambda}^{(0)}$, and $\beta_j^{(0)} = 0$ for all j. The parameters are then updated in the order $\boldsymbol{\beta}$, $\boldsymbol{\omega}$, τ, and $\boldsymbol{\lambda}$. In particular, the Gibbs sampler draws a state $\boldsymbol{\beta}^{(1)}$ from its



(a) Under the prior preconditioner.

(b) Under the Jacobi preconditioner.

Fig. S6. Histograms of the eigenvalues of the preconditioned matrices as in Figure 3.2. The only differences here are that 1) the preconditioned matrices are based on a posterior sample from the treatment effect model (5.25) and 2) the trimmed version for the Jacobi preconditioner removes the eigenvalues in the range $[-0.5, 0.5]$ as this choice better demonstrates the tail behavior here.

posterior distribution conditional on $\boldsymbol{\omega}^{(0)}$, $\tau^{(0)}$, and $\boldsymbol{\lambda}^{(0)}$. With this initialization, all the values $\beta_j^{(1)}$'s end up being shrunk close to zero compared with their actual posterior means. The subsequent update of τ conditional on $\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\omega}^{(1)}$ (with $\boldsymbol{\lambda}$ marginalized out) yields $\tau^{(1)} \approx 10^{-3}$ as can be seen Figure S4. In particular, while we initialize the global shrinkage parameter value as 0.01, its value shoots down to 10^{-3} after one iteration of the Gibbs sampler. This is caused by the most components of $\boldsymbol{\beta}$ initially shrunken to 0, which in turn is caused by the choice $\tau^{(0)} = 0.01$ and $\lambda_j^{(0)} = 1$.

A more careful initialization of τ and $\boldsymbol{\lambda}$, or alternatively of $\boldsymbol{\beta}$, is likely to speed up the convergence of the Markov chain and is worth future investigations. Assessing the relative importance of β_j through pre-screening techniques, such as those based on marginal correlations, may be used to initialize $\tau\lambda_j$'s.

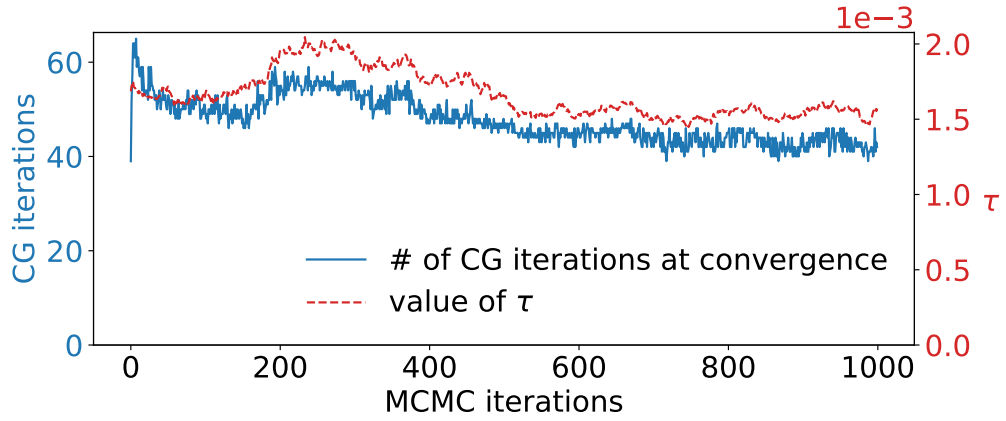


Fig. S7. Plot as in Figure S4 of the number of CG iterations required during each update of β by the CG-accelerated Gibbs sampler for the treatment effect model (5.25).

S4. Problem with approximating $\tilde{\Phi}$ by thresholding of $\tau\lambda_j$'s

In Section 2.4, we observed that the (i, j)-th entry of $\tau^2\boldsymbol{\Lambda}\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}\boldsymbol{\Lambda}$ is small whenever $\tau\lambda_i \approx 0$ or $\tau\lambda_j \approx 0$. Given this observation, one may wonder if we can obtain a convenient low-rank approximation of the prior-preconditioned matrix $\tilde{\boldsymbol{\Phi}} = \tau^2\boldsymbol{\Lambda}\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}\boldsymbol{\Lambda} + \boldsymbol{I}_p$ by zeroing out $\tau\lambda_j$'s at some threshold. This is *not* the case in general as we will show now. Intuitively, the problem is as follows: while the ordered local shrinkage parameter $\lambda_{(k)}$ decays reasonably quickly as k increases, there is no clear "gap" where $\lambda_{(k+1)} \ll \lambda_{(k)}$. For example, the histogram of a posterior draw of $\tau\lambda_j$'s in Figure 3.4 shows clearly that there is no such gaps.

We can assess the quality and utility of the thresholding approximation by using it as a pre-conditioner for CG in solving the system $\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\beta}} = \tilde{\boldsymbol{b}}$. In other words, we consider using the thresholding approximation on top of the prior preconditioning. Let $(\tau^2\boldsymbol{\Lambda}\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}\boldsymbol{\Lambda})_{(k)}$ denote a matrix obtained by thresholding the entries of $\tau^2\boldsymbol{\Lambda}\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}\boldsymbol{\Lambda}$ to zero except for the $k \times k$ block corresponding to the k largest local shrinkage parameters $\lambda_{(1)}, \ldots, \lambda_{(k)}$. Then the thresholding approximation

$$\tilde{\boldsymbol{\Phi}}_{(k)} = (\tau^2\boldsymbol{\Lambda}\boldsymbol{X}^\intercal\boldsymbol{\Omega}\boldsymbol{X}\boldsymbol{\Lambda})_{(k)} + \boldsymbol{I}_p \tag{S6}$$

is the identity perturbed by the $k \times k$ block along the diagonal. As such, using it as a preconditioner requires the one time cost of computing and factorizing the $k \times k$ block, which requires $O(k^2n + k^3)$ arithmetic operations. The quality of the approximation should improve as k increases but so does the computational cost. In particular, at some point the $O(k^2n + k^3)$ cost of preparing the thresholding preconditioner overwhelms the $O(np)$ cost of each CG iteration and becomes the computational bottleneck. When an adequate approximation requires such a large k, therefore, there is no benefit of using the thresholding approximation.

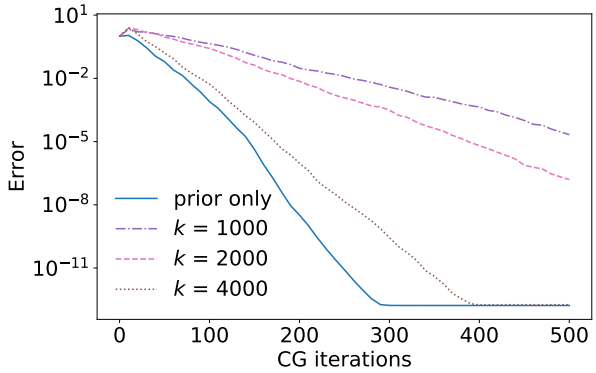


Fig. S8. Plot of the CG approximation errors (with the same error metric as used in Figure 3.1) as a function of the number of CG iterations. The CG sampler here is applied to the conditional distribution of $\boldsymbol{\beta}$ arising from the simulated data example with 10 signals as described in Section 3. Each line corresponds to a different threshold level k for the approximation (S6). The blue line labeled 'prior only' corresponds to CG applied to the prior preconditioned system without any preconditioner.

We use the example of Section 3.3 to study the effects of preconditioning the system $\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\beta}} = \tilde{\boldsymbol{b}}$ with the thresholding approximation $\tilde{\boldsymbol{\Phi}}_{(k)}$. As before, the CG sampler is applied to the distribution (1.3) arising from the simulated data with 10 signals out of $p = 10{,}000$ predictors. Figure S8 shows the results for $k = 1{,}000, 2{,}000,$ and $4{,}000$. The convergence rates of these preconditioned CG iterations are compared to that of the CG iterations applied to the prior-preconditioned system without any additional preconditioning. If a preconditioner is a good approximation of $\tilde{\boldsymbol{\Phi}}$, the preconditioned CG should yield convergence in a very small number of iterations — for example, a perfect approximation would induce the convergence after one iteration. It is clear from Figure S8, however, that the thresholding approximation does more harm than good in terms of the CG convergence rate, especially when k is taken small relative to the size of $\tilde{\boldsymbol{\Phi}}$. We can therefore conclude that the thresholding strategy yields a poor approximation except when k starts to become almost as large as p.

We repeated the same experiments on the thresholding approximation $\tilde{\boldsymbol{\Phi}}_{(k)}$ using the other posterior distributions discussed in the manuscript. Across the experiments, no computational gain could be achieved by using the thresholding approximation as a preconditioner. More precisely, to yield a good enough approximation, the value of k had to be so large that preparing the preconditioner itself became a computational bottleneck. The convergence rates of CG preconditioned by the thresholding approximation are shown in Figure S9 for the dabigatran and warfarin comparison examples of Section 5.

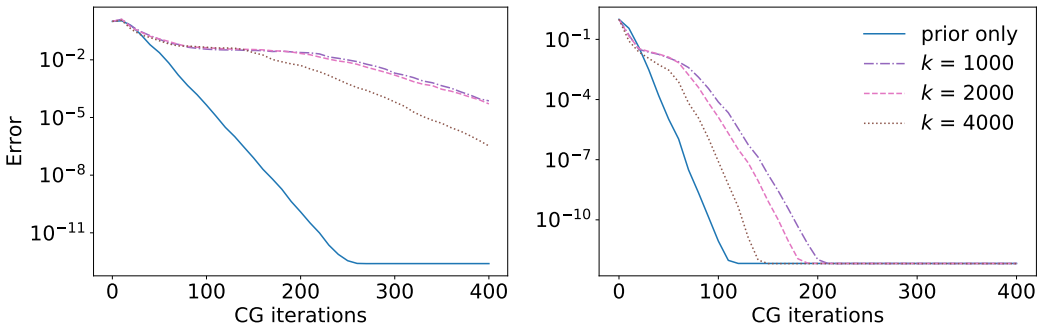


Fig. S9. Plots of the CG approximation errors (with the same error metric as used in Figure 3.1) as a function of the number of CG iterations. The CG samplers are applied to the conditional distribution of β arising from the propensity score model (5.24) (the left plot) and treatment effect model (5.25) (the right plot).

References for Supplement

Demmel, J. W. (1997) *Applied Numerical Linear Algebra*, vol. 56. Society for Industrial and Applied Mathematics.

Driscoll, T. A., Toh, K.-C. and Trefethen, L. N. (1998) From potential theory to matrix iterations in six steps. *SIAM review*, **40**, 547–578.

Golub, G. H. and Van Loan, C. F. (2012) *Matrix computations*, vol. 3. Johns Hopkins University Press.

Greenbaum, A. (1979) Comparison of splittings used with the conjugate gradient algorithm. *Numerische Mathematik*, **33**, 181–193.

Horn, R. A. and Johnson, C. R. (2012) *Matrix Analysis*. Cambridge University Press.

Kuijlaars, A. B. J. (2006) Convergence analysis of Krylov subspace iterations with methods from potential theory. *SIAM review*, **48**, 3–40.

Meurant, G. A. (2006) *The Lanczos and Conjugate Gradient Algorithms: from Theory to Finite Precision Computations*. Society for Industrial and Applied Mathematics.

Saad, Y. (2003) *Iterative Methods for Sparse Linear Systems*, vol. 82. Society for Industrial and Applied Mathematics.

— (2011) *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics.

Trefethen, L. N. and Bau, D. (1997) *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics.

Van der Vorst, H. A. (2003) *Iterative Krylov Methods for Large Linear Systems*, vol. 13. Cambridge University Press.