Check for updates

*The Canadian Journal of Statistics*
Vol. 49, No. 2, 2021, Pages 421–437
*La revue canadienne de statistique*



# Weighted Bayesian bootstrap for scalable posterior distributions

Michael A. NEWTON[1,2*], Nicholas G. POLSON[3], and Jianeng XU[3]

[1]*Department of Statistics, University of Wisconsin-Madison, Madison, WI, U.S.A.*
[2]*Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, Madison, WI, U.S.A.*
[3]*Booth School of Business, University of Chicago, Chicago, IL, U.S.A.*

*Key words and phrases:* Deep learning; Markov chain Monte Carlo; regularization; trend filtering; weighted bootstrap.

*MSC 2010:* Primary 62F40; secondary 62F15.

*Abstract:* We introduce and develop a weighted Bayesian bootstrap (WBB) for machine learning and statistics. WBB provides uncertainty quantification by sampling from a high dimensional posterior distribution. WBB is computationally fast and scalable using only off-the-shelf optimization software. First-order asymptotic analysis provides a theoretical justification under suitable regularity conditions on the statistical model. We illustrate the proposed methodology in regularized regression, trend filtering and deep learning and conclude with directions for future research. *The Canadian Journal of Statistics* 49: 421–437; 2021 © 2020 Statistical Society of Canada

*Résumé:* Les auteurs développent un bootstrap pondéré bayésien (BPB) pour l'apprentissage machine et la statistique. Le BPB offre une quantification de l'incertitude en échantillonnant à partir d'une loi a posteriori en haute dimension. Il est rapide à calculer et peut être mis à l'échelle en utilisant uniquement des logiciels d'optimisation prêts à l'emploi. Les auteurs justifient la méthode théoriquement à l'aide d'une analyse asymptotique du premier ordre sous des conditions de régularité du modèle statistique. Ils illustrent la méthodologie proposée avec des modèles de régression régularisée, de filtrage des tendances, et d'apprentissage profond. Ils concluent par des pistes de recherche. *La revue canadienne de statistique* 49: 421–437; 2021 © 2020 Société statistique du Canada

## 1. INTRODUCTION

Weighted Bayesian bootstrap (WBB) is a simulation-based algorithm for assessing uncertainty in machine learning and statistics. Uncertainty quantification (UQ) is an active area of research, particularly in high-dimensional inference problems (Wang & Swiler, 2018). Whilst there are computationally fast and scalable algorithms for training models in a wide variety of contexts, uncertainty assessments are still required, as are methods to compute these assessments. Bayesian analysis offers a general solution, but developing computationally fast scalable algorithms for sampling a posterior distribution is a notoriously hard problem. WBB makes a contribution to this literature by showing how off-the-shelf optimization algorithms, such as convex optimization or stochastic gradient descent (SGD), can be adapted to provide uncertainty assessments. Our goal

Additional Supporting Information may be found in the online version of this article at the publisher's website.
*Author to whom correspondence may be addressed.
*E-mail: newton@stat.wisc.edu*

© 2020 The Authors. The Canadian Journal of Statistics/La revue canadienne de statistique
published by Wiley Periodicals LLC on behalf of Statistical Society of Canada.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

here is to marry Bayesian uncertainty techniques with state-of-the-art optimization methods and software systems.

For relatively simple statistical models, the weighted likelihood bootstrap (WLB) method provides approximate posterior sampling through repeated optimization of a randomly weighted likelihood function (Newton & Raftery, 1994). The proposed WBB extends the WLB to a broad class of contemporary statistical models by leveraging advances in optimization methodology. Essentially, the WLB used optimization of certain randomized objective functions to enable approximate marginalization (i.e., integration) required in Bayesian analysis. The same idea—optimize a randomized objective function to achieve posterior sampling—is at the heart of the proposed WBB method, though some changes to the WLB procedure are required to carry out this program for the models considered. Theoretical support for the WLB approximation is based on connections between posterior variation and curvature of the log-likelihood revealed through repeated optimization of randomly weighted likelihoods. By contrast, the proposed WBB calculates a series of randomized posterior modes rather than randomized likelihood maximizers. A key rationale for this proposal is that high dimensional posterior modes are now readily computable, thanks to systems such as `TensorFlow` (Abadi et al., 2015) and `Keras` (Chollet, 2015) that deploy SGD and convex optimization methods for large-scale problems, such as on neural network architectures used in deep learning (LeCun, Bengio & Hinton, 2015). By linking random weighting with advanced optimization, we expose a simple scheme for approximate uncertainty quantification in a wide class of statistical models.

Quantifying uncertainty is typically unavailable in a purely regularization optimization method. We contend that UQ is available directly by repeated optimization of randomized objective functions, using the same computational tools that produce the primary estimate, rather than through Markov chain Monte Carlo, variational methods, approximate Bayesian computation (ABC), or other techniques. See Green et al. (2015) for a good summary of Bayesian computation history. Thus, uncertainty assessments are provided at little extra effort over the original training computations. A further benefit is that with extra computational cost, it is straightforward to add a regularization path across hyper-parameters (e.g., simply repeat WBB on different $\lambda$), which is usually difficult to compute in traditional Bayesian sensitivity analysis. We use predictive cross-validation techniques in this regard.

The rest of the article is outlined as follows. Section 2 develops our weighted Bayesian Bootstrap (WBB) algorithm. Section 3 provides applications to high dimensional sparse regression, trend filtering and deep learning. WBB can also be applied to Bayesian tree models (Taddy et al., 2015). Section 4 indicates several directions for future research, including bootstrap filters in state-space models (Gordon, Salmond & Smith, 1993) and connections to the resampling-sampling perspective in sequential Bayesian inference (Lopes, Polson, & Carvalho, 2012).

## 2. WEIGHTED BAYESIAN BOOTSTRAP

### 2.1. Setting

We work with a broad class of statistical models for data structures involving outcomes and covariates. Examples considered in Section 3 include regression, trend-filtering, and deep learning. Let $y = (y_1, y_2, \dots, y_n)$ be an $n$-vector of outcomes and let $\theta = (\theta_1, \theta_2, \dots, \theta_p)$ be a $p$-dimensional parameter of interest. Covariate data may be organized in an $n \times p$ matrix $A$ whose rows are the design points (or "features") $a_i^T$ where we index observations by $i$ and parameters by $j$. A large number of estimation/training problems can be expressed in the form

$$\underset{\theta \in \mathcal{R}^d}{\text{minimize}} \quad \mathcal{L}(\theta) := l(y|\theta) + \lambda \phi(\theta), \tag{1}$$

 DOI: 10.1002/cjs.11570

where $l(y|\theta) = \sum_{i=1}^{n} l_i(y_i|\theta)$ is a measure of fit (or "empirical risk function") depending on $\theta$ and $y$ and implicitly on $A$. The penalty function, or regularization term, $\lambda\phi(\theta)$, may encode soft or hard constraints, and is controlled by a hyper-parameter, $\lambda > 0$, whose values index an entire path of solutions. The penalty function $\phi(\theta)$ effects a favourable bias-variance estimation trade-off and provides extensive modelling flexibility (Wellner & Zhang, 2012). To accommodate contemporary applications we allow $\phi(\theta)$ to have points in its domain where it fails to be differentiable (e.g., $L^1$ norm). If we treat data, $y$, as arising from a probabilistic model parameterized by $\theta$, then the likelihood function $p(y|\theta)$ yields the model-associated measure of fit $l(y|\theta) = -\log p(y|\theta)$. The maximum likelihood estimator (MLE) is $\hat{\theta} := \text{argmax}_\theta\, p(y|\theta)$, though of course this usually differs from the solution to Equation (1): $\theta^* := \arg\min\{l(y|\theta) + \lambda\phi(\theta)\}]$. We recall a key duality between regularization and Bayesian analysis.

### 2.2. Bayesian regularization duality

From the Bayesian perspective, the measure of fit, $l(y|\theta) = -\log p(y|\theta)$, and the penalty function, $\lambda\phi(\theta)$, correspond to the negative logarithms of the likelihood and prior distribution in the model

$$\begin{aligned} p(y|\theta) &\propto \exp\{-l(y|\theta)\}\ , \\ p(\theta) &\propto \exp\{-\lambda\phi(\theta)\}, \\ p(\theta|y) &\propto \exp\{-(l(y|\theta) + \lambda\phi(\theta))\}. \end{aligned} \tag{2}$$

This posterior $p(\theta|y)$ is often a proper distribution over $\mathcal{R}^d$, even if the prior $p(\theta)$ is not proper. The well-known equivalence between regularization and Bayesian methods is seen, for example, in regression with a Gaussian regression model subject to a penalty such as an $L^2$-norm (ridge) Gaussian prior or $L^1$-norm (LASSO) double exponential prior. By this duality, the posterior mode, or maximum a posteriori (MAP) estimate, is $\theta^*$, a solution to Equation (1). See Gribonval & Machart (2013) for a nuanced view of the connection between Equations (1) and (2) in Gaussian regression models.

### 2.3. Optimization

Advances in optimization methodology provide efficient algorithms to compute $\theta^* = \arg\min \mathcal{L}(\theta)$ for a wide range of loss and penalty functions. Theory is well developed in the case of convex objective functions (e.g., Bertsekas, Nedi & Ozdaglar, 2003; Boyd & Vandenberghe, 2004). For example if loss $l$ is convex and differentiable in $\theta$ and penalty $\phi$ is convex, then a necessary and sufficient condition for $\theta^*$ to minimize $l(y|\theta) + \lambda\phi(\theta)$ is

$$0 \in \partial\left\{l(y|\theta^*) + \lambda\phi(\theta^*)\right\} = \nabla l(y|\theta^*) + \lambda\partial\phi(\theta^*), \tag{3}$$

where $\partial$ is the subdifferential operator (the set of subgradients of the objective), in this case the sum of a point and a set. Though not a formula for $\theta^*$, such as given by the normal equations in linear regression, Equation (3) usefully guides algorithms that aim to solve $\theta^*$. For example, under separability conditions on the penalty function, coordinate descent algorithms effectively solve for $\theta^*$; see Wright (2015), or Hastie, Tibshirani & Wainwright (2015, chap. 5) for a statistical perspective. The optimization literature also characterizes $\theta^*$ as the fixed point of a proximal operator $prox_{\gamma\mathcal{L}}(\theta) = \arg\min_z\{\mathcal{L}(z) - \frac{1}{2\gamma}\|z - \theta\|_2^2\}$, which opens the door to powerful MM algorithms and related schemes; see Lange (2016, chap. 5), Polson & Scott (2015), and Polson, Scott & Willard (2015). Beyond convexity, the guarantees are weaker (e.g., local not global minima) and the algorithms are many (e.g., Nocedal & Wright, 2006). Gradient descent or SGD are effective in many cases, owing to parameter dimensionality and structure of the gradients. The appendix develops SGD for one example.

Advances in applied optimization provide effective software tools for data analysis. For example, the R package `glmnet` deploys coordinate descent for loss functions arising from generalized linear models and LASSO or elastic net penalties (Friedman, Hastie & Tibshirani, 2010). To solve the generalized LASSO problem, the R package `genlasso` deploys a dual path algorithm (Arnold & Tibshirani, 2014). A variety of general purpose optimization tools for statistics are compiled at the optimization view at CRAN (`https://cran.r-project.org/web/views/Optimization.html`). For machine learning, the `TensorFlow` system has greatly simplified gradient descent, SGD, and related algorithms for many applications (Abadi et al., 2015).

### 2.4. WBB Algorithm

We now define the WBB. Recalling the original objective function Equation (1), we form the randomized objective

$$\mathcal{L}_{\mathbf{w}}(\theta) = \left\{ \sum_{i=1}^{n} w_i l_i(y_i|\theta) \right\} + \lambda w_0 \phi(\theta), \tag{4}$$

where entries of $\mathbf{w} = (w_0, w_1, \ldots, w_n)$ are independent and identically distributed (i.i.d.) standard exponentially distributed random weights, generated by the analyst and independently from the data $y$. Equivalently, $w_i = \log(1/u_i)$ where $u_i$'s are i.i.d. Uniform$(0, 1)$. When $p$ is relatively large and $\phi(\theta)$ is separable as $\phi(\theta) = \sum_{j=1}^{p} \phi_j(\theta_j)$, we recommend an extension in which $w_0 = (w_{0,1}, w_{0,2}, \ldots, w_{0,p})$ allows separate i.i.d. random weights on the prior regularization terms $\phi_j(\theta_j)$, namely $\lambda \sum_{j=1}^{p} w_{0,j} \phi_j(\theta_j)$, to help with sparsity and to reduce the effect of occasionally large scalar weight. In either case, associated with any vector $\mathbf{w}$ is the solution, $\theta^*_{\mathbf{w}} = \arg\min \mathcal{L}_{\mathbf{w}}(\theta)$. Our basic conjecture is that the conditional distribution of $\theta^*_{\mathbf{w}}$—the distribution induced by $\mathbf{w}$ with the data fixed—approximates the Bayesian posterior Equation (2). For any measurable set $\mathcal{B}$ in the parameter space,

$$\Pr\left(\theta^*_{\mathbf{w}} \in \mathcal{B}|y\right) \approx \int_{\mathcal{B}} p(\theta|y)\, d\theta. \tag{5}$$

Section 2.5 provides an asymptotic argument in support of Equation (5), and we investigate the approximation numerically in a few examples in Section 3. Assuming this conjecture is true, we have a straightforward optimization-based algorithm for approximate posterior sampling:

**Algorithm 1** Weighted Bayesian bootstrap

**Input:**
- data: $\mathcal{D} = (y, A)$
- model structure: $\mathcal{M} = \left(\{l_i\}, \lambda, \phi\right)$
- number of draws: $T$

**Output:** $T$ parameter samples $\{\theta^{*,t}\}$

Function **WBB**$(\mathcal{D}, \mathcal{M}, T)$:
**for all** $t = 1$ to $T$ **do**
- Realize: $(u_1, \ldots, u_n) \sim_{i.i.d.}$ Uniform$(0, 1)$
- Construct: $w_i \leftarrow \log(1/u_i), \forall i$.
- Independently construct $w_0$ as either a univariate Exponential (common weight case) or as a vector of i.i.d. Exponentials (separate weights case)
- Set $\mathbf{w} = (w_0, w_1, \ldots, w_n)$
- Compute: $\theta^{*,t} \leftarrow \arg\min \mathcal{L}_{\mathbf{w}}(\theta)$

**end for**

When optimization on the original problem Equation (1) is fast and scalable, so too is the WBB. Weights $\mathbf{w}$ and the corresponding $\theta^*_{\mathbf{w}}$ are independent across $t$, making it possible to speed up the algorithm via parallel computing. To choose the amount of regularization $\lambda$ which is assumed to be fixed for all sets of $\mathbf{w}$, we can use the marginal likelihood $m_\lambda(y)$, estimated by bridge sampling (Gelman & Meng, 1998) or simply using predictive cross-validation. Next we consider the approximation Equation (5) from an asymptotic perspective.

### 2.5. WBB Properties

Conditions under which the target posterior distribution Equation (2) is approximately Gaussian are well established (Kleijn & van der Vaart, 2012). For example, when data form a random sample from fixed distribution $p(y_i|\theta_0)$ that resides in a sufficiently regular model, and when the prior is smooth and positive around $\theta_0 \in \mathcal{R}^p$, we have

$$\theta|\, y \sim_{\text{approx}} N_p\left(\theta^*_n, J_n^{-1}(\theta^*_n)\right), \tag{6}$$

where including sample size $n$ as an explicit subscript, we have posterior mode $\theta^*_n = \arg\min \mathcal{L}_n(\theta)$, and where $J_n(\theta^*_n) = nj(\theta^*_n)$ is the Fisher information matrix evaluated at $\theta^*_n$. Here $j(\theta)$ is the information per sample, and $y = (y_1, \ldots, y_n)$ denotes data. Johnstone (2010) studies Bernstein–von Mises theorem in high dimensional settings where $p$ grows with $n$ and the situation is very different. Centring on the posterior mode, rather than the MLE, improves accuracy in many cases.

As to the WBB distribution, consider a one-term Taylor expansion of $\nabla\mathcal{L}_{\mathbf{w},n}(\theta)$ about the posterior mode $\theta^*_n$, which is allowable for sufficiently smooth loss and penalty terms:

$$\nabla\mathcal{L}_{\mathbf{w},n}(\theta) = \nabla\mathcal{L}_{\mathbf{w},n}(\theta^*_n) + \nabla^2\mathcal{L}_{\mathbf{w},n}(\theta^*_n)(\theta - \theta^*_n) + R_n, \tag{7}$$

where $R_n$ is an error term and $\nabla$ and $\nabla^2$ record the gradient vector and matrix of second partial derivatives, respectively, of the weighted objective function. Evaluating this expansion at $\theta^*_{\mathbf{w},n} = \arg\min \mathcal{L}_{\mathbf{w},n}(\theta)$ zeros out the left hand side of Equation (7) and leads to:

$$\sqrt{n}\left(\theta^*_{\mathbf{w},n} - \theta^*_n\right) = -\left(\frac{1}{n}\nabla^2\mathcal{L}_{\mathbf{w},n}(\theta^*_n)\right)^{-1}\left(\frac{1}{\sqrt{n}}\nabla\mathcal{L}_{\mathbf{w},n}(\theta^*_n)\right) + \tilde{R}_n, \tag{8}$$

where $\tilde{R}_n$ is another error term. Following Newton & Raftery (1994), we recognize that the $\mathbf{w}$-induced variation in Equation (8), conditional upon the data, causes the matrix factor to be approximately the inverse information $[J_n(\theta^*_n)]^{-1}$, the score-like second factor to be approximately mean-zero Gaussian with covariance equal to $J_n(\theta^*_n)$, and the error $\tilde{R}_n$ to be negligible. Thus, compared to the target posterior variation Equation (6), we have WBB variation:

$$\theta^*_{\mathbf{w},n}\Big|\, y \sim_{\text{approx}} N_p\left(\theta^*_n, J_n^{-1}(\theta^*_n)\right).$$

In a relatively narrow asymptotic sense, therefore, the WBB procedure is approximating the target posterior distribution, as both are approximately Gaussian with the same mean and covariance. Details of the asymptotic analysis follow the WLB case presented in Newton & Raftery (1994), and differ only slightly in our use of the posterior mode $\theta^*_n$ in place of the MLE $\hat{\theta}_n$, and also in our incorporation of weight $w_0$ on the penalty term of the objective function. At this level of first-order asymptotic analysis, neither of these features affects the limiting conditional Gaussian distribution of $\theta^*_{\mathbf{w},n}$.

We note that rescaling in Equation (4) has no effect on solutions $\theta^*_{\mathbf{w},n}$, and so it is equivalent in the construction to use normalized weights $\tilde{\mathbf{w}}$ that sum to unity. Such $\tilde{\mathbf{w}}$ are uniformly distributed over the unit simplex, and thus correspond to a specific Dirichlet distribution. Exponential/Dirichlet weights are motivated from the perspective of both inference, related to the original Bayesian bootstrap (Rubin, 1981; Muliere & Secchi, 1996), and computation, owing to benefits of smoothly varying weights. Other weight distributions may also be effective (Barbe & Bertail, 2012).

We aim to use WBB samples as approximate posterior samples for uncertainty quantification. It remains unknown in general what is the relationship between this WBB distribution and the posterior distribution associated with any specific prior. The first-order asymptotic approximation above is constructed so that the prior structure is asymptotically negligible; one could say, then, that WBB approximates any one of a number of different posterior distributions. In particular, the WBB samples may not provide a close approximation to the posterior formed from the same prior as used in the penalty term in the objective function. We deploy numerical experiments to understand the approximation better in finite samples.

## 3. NUMERICAL EXPERIMENTS

We illustrate the proposed methodology with a number of scenarios to assess the quality of the WBB approximation.

### 3.1. LASSO Experiment

First, consider a simple univariate normal means problem with a LASSO prior where

$$y|\theta \sim N(\theta, 1^2), \quad \theta \sim Laplace(0, 1/\lambda).$$

Given the i.i.d. exponential weights $w_1$ and $w_0$, the weighted posterior mode $\theta^*_{\mathbf{w}}$ is given by

$$\theta^*_{\mathbf{w}} = \arg\min_{\theta\in\Theta} \left\{ \frac{w_1}{2}(y-\theta)^2 + \lambda w_0|\theta| \right\}.$$

This is sufficiently simple for an exact WBB solution in terms of the soft thresholding proximal operator:

$$\theta^*_{\mathbf{w}} = \begin{cases} y - \lambda w_0/w_1 & \text{if } y > \lambda w_0/w_1, \\ y + \lambda w_0/w_1 & \text{if } y < -\lambda w_0/w_1, \\ 0 & \text{if } |y| \le \lambda w_0/w_1. \end{cases}$$

The WBB mean $E_{\mathbf{w}}(\theta^*_{\mathbf{w}}|y)$ is approximated by the sample mean of $\{\theta^{*,t}_{\mathbf{w}}\}_{t=1}^T$. On the other hand, Mitchell (1994) gives the expression for the posterior mean,

$$\begin{aligned} E(\theta|y) &= \frac{\int_{-\infty}^{\infty} \theta \exp\left\{-(y-\theta)^2/2 - \lambda|\theta|\right\} d\theta}{\int_{-\infty}^{\infty} \exp\left\{-(y-\theta)^2/2 - \lambda|\theta|\right\} d\theta} \\ &= \frac{F(y)}{F(y)+F(-y)}(y+\lambda) + \frac{F(-y)}{F(y)+F(-y)}(y-\lambda) \\ &= y + \frac{F(y)-F(-y)}{F(y)+F(-y)}\lambda, \end{aligned}$$

where $F(y) = \exp(y)\Phi(-y-\lambda)$ and $\Phi(\cdot)$ is the c.d.f. of standard normal distribution. We plot the WBB mean versus the exact posterior mean in Figure 1. Interestingly, the WBB algorithm

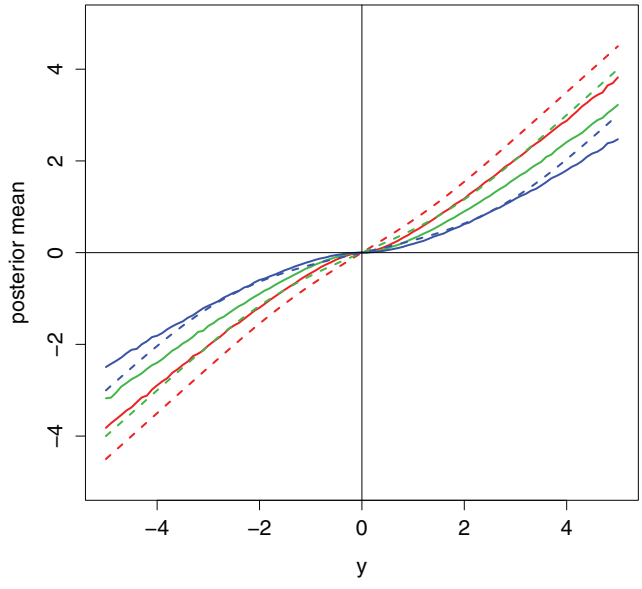


FIGURE 1: Normal means model with LASSO prior: WBB mean $E_{\mathbf{w}}(\theta^*_{\mathbf{w}}|y)$ (in solid lines) versus exact posterior mean $E(\theta|y)$ (in dashed lines).

shrinks the posterior means towards zero. WBB samples are approximate posterior draws, though the algorithm structure does not permit a clear description of the prior that is in force for this sampled distribution. Whatever is this effective prior, it may be different from the prior encoded in the penalty used in repeated optimization, as the result in Figure 1 suggests.

A simulation study is conducted in order to further investigate WBB in the context of regression. We simulate data from the linear model,

$$y = X\beta + \epsilon,\ \epsilon \sim N(0, \sigma^2_\epsilon I),$$

where $X$ is $n \times p$ and $y$ is $n \times 1$. The design matrix $X$ is generated by drawing its $n$ rows independently from a $p$-dimensional normal distribution, $N(\mathbf{0}, \Sigma)$. Within each row, the covariance between entries in columns $i$ and $j$ is set to be $\Sigma_{i,j} = 0.1 \times 0.8^{|i-j|}$. Furthermore, we set the noise variance $\sigma^2_\epsilon = \|X\beta\|^2_2/(2n)$. In this setting, the signal-to-noise ratio is 2. Figure 2 displays WBB samples (kernel density estimates) and posteriors via MCMC in one of the simulation settings; $n = 50, p = 2, \sigma^2_\epsilon = 1, \Sigma = [[1, 0.3]', [0.3, 1]']$ and $\beta_1 = 0, \beta_2 = 2$. The WBB (on the $L^1$, LASSO prior penalty) is compared to posteriors computed via MCMC under several different priors: the same $L^1$ prior encoded in the WBB penalty, and also two $L^0$ penalties, $p(\beta) \propto \exp\{-\lambda\|\beta\|_0\}$ and $p(\beta) \propto \exp\{-5\lambda\|\beta\|_0\}$ (Polson & Sun, 2019). The WBB samples entail a spike at $\beta_1 = 0$, indicating a positive probability mass in the effective prior. No such posterior mass is evident in the $L^0$ priors nor possible in the $L^1$ prior.

Next we use the simulation to compare WBB distributions to MCMC-computed posteriors in a range of regression settings. We consider both sparse and dense cases for the coefficient vector $\beta = [\beta_1, \beta_2, \ldots, \beta_p]'$:

A(i). $\beta_j = 1$ for $1 \le j \le 10$ and $\beta_j = 0$ for $j > 10$.
A(ii). $\beta_j = 1$ for $1 \le j \le 5$, $\beta_j = 10$ for $6 \le j \le 10$ and $\beta_j = 0$ for $j > 10$.
B. $\beta_j = 1$ for $1 \le j \le p$.

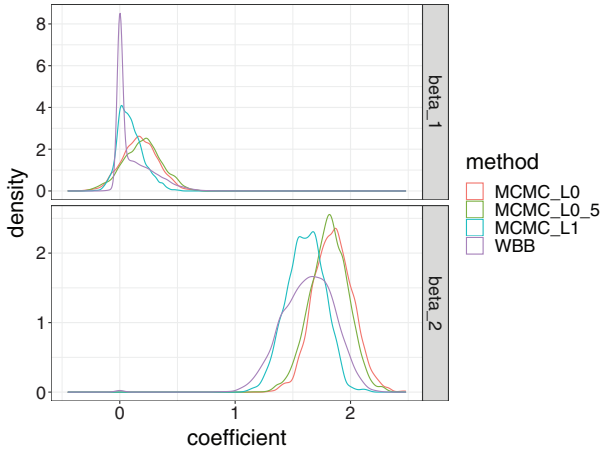


FIGURE 2: WBB samples under LASSO penalty, compared with posteriors under LASSO prior and $L^0$ prior computed by MCMC.

For each setting of $\beta$, we consider $p = 40, 60, 80, 100, 120$. To investigate the performance of WBB under different sample sizes, we also choose two settings of $n$: $n = 100$ for all $p$, or $n = p/2$ for all $p$.

We compare the WBB with LASSO prior and separate weights to Markov chain Monte Carlo under the Bayesian LASSO (Park & Casella, 2008; Carlin & Polson, 1991). The two methods entail different prior penalties:

$$\text{WBB}: \quad \lambda\phi(\beta|\mathbf{w}) = \lambda \sum_{j=1}^{p} w_{0,j}|\beta_j|,$$

$$\text{BLASSO}: \quad \phi(\beta|\sigma^2) = -\log\left(\frac{\lambda}{2\sigma}\right) + \frac{\lambda}{\sigma}\sum_{j=1}^{p}|\beta_j|.$$

Bayesian LASSO imposes a noninformative marginal prior on $\sigma^2$, $\pi(\sigma^2) = 1/\sigma^2$ and the posterior distribution is sampled by a Gibbs sampler, with $\lambda$ chosen by maximizing the marginal likelihood. In WBB, $\lambda$ is chosen via cross-validation, using standard unweighted LASSO. Here the original LASSO prior is separable and we study the separate-weights version of WBB. In high-dimensional cases, a large common $w_0$ multiplied to $\phi(\beta)$ can introduce extra sparsity to all marginal posteriors. We use separate weights in an effort to overcome this issue. For comparison criteria, we present the estimation MSE of coefficients $\beta$ (use posterior mean as our estimate), out-of-sample prediction MSE (test sets are of the same size as the corresponding training sets), and 95% credible interval coverage. Fixing $(p, n, \beta)$, we draw $T = 200$ posterior samples by each method and the estimation procedure is repeated over $B = 500$ independent datasets $\{(X, y)^{(b)}\}_{b=1}^{B}$. Let $\beta^{*,t}(b)$ denote the $t$th posterior draw, with respect to the $b$th dataset. For each coordinate $j$, coverage is calculated using $\{\beta_j^{*,t}(b) : 1 \le t \le T, 1 \le b \le B\}$. The coverages are then averaged across $1 \le j \le p$.

The complete results for estimation error, out-of-sample prediction error, and credible interval coverage are displayed in Tables 1–3, respectively. In almost all cases when $p \ge 60$, WBB has lower estimation error than BLASSO. WBB shows comparable or superior out-of-sample prediction in most cases. Credible intervals of neither WBB nor BLASSO have exact 95% coverage. WBB performs well when $p \ge 100$ and $\beta$ is sparse, though it often under covers. When $\beta_j = 1$ for all $1 \le j \le p$, BLASSO intervals are too wide while WBB intervals have coverage close to 95% when $40 \le p \le 80$. Though limited in scope, this simulation reveals some distinctions and also broad similarities between WBB and Bayesian LASSO in both posterior

 DOI: 10.1002/cjs.11570

TABLE 1: Coefficient estimation mean squared error (MSE).

| | | $p$ | 40 | 60 | 80 | 100 | 120 |
|---|---|---|---|---|---|---|---|
| $n = 50$ | A(i) | BLASSO | 0.13 | 0.09 | 0.06 | 0.05 | 0.04 |
| | | WBB | 0.19 | 0.06 | 0.05 | 0.04 | 0.03 |
| | A(ii) | BLASSO | 5.59 | 3.74 | 2.90 | 2.33 | 1.96 |
| | | WBB | 7.40 | 2.89 | 2.31 | 1.90 | 1.62 |
| | B | BLASSO | 0.67 | 0.67 | 0.68 | 0.70 | 0.70 |
| | | WBB | 1.77 | 0.49 | 0.50 | 0.51 | 0.52 |
| $n = p/2$ | A(i) | BLASSO | 0.13 | 0.08 | 0.06 | — | 0.05 |
| | | WBB | 0.14 | 0.08 | 0.05 | — | 0.03 |
| | A(ii) | BLASSO | 6.88 | 4.09 | 2.88 | — | 1.94 |
| | | WBB | 6.58 | 3.80 | 2.53 | — | 1.47 |
| | B | BLASSO | 0.52 | 0.56 | 0.64 | — | 0.74 |
| | | WBB | 0.68 | 0.62 | 0.55 | — | 0.48 |

TABLE 2: Out-of-sample prediction MSE.

| | | $p$ | 40 | 60 | 80 | 100 | 120 |
|---|---|---|---|---|---|---|---|
| $n = 50$ | A(i) | BLASSO | 3.35 | 3.42 | 3.43 | 3.61 | 3.68 |
| | | WBB | 3.34 | 3.37 | 3.40 | 3.60 | 3.72 |
| | A(ii) | BLASSO | 119.65 | 124.03 | 130.49 | 132.21 | 135.46 |
| | | WBB | 121.45 | 123.40 | 129.89 | 134.04 | 140.44 |
| | B | BLASSO | 20.24 | 33.18 | 46.75 | 61.86 | 80.57 |
| | | WBB | 20.61 | 32.47 | 46.27 | 60.59 | 78.71 |
| $n = p/2$ | A(i) | BLASSO | 4.55 | 4.15 | 3.83 | — | 3.61 |
| | | WBB | 3.65 | 3.75 | 3.66 | — | 3.65 |
| | A(ii) | BLASSO | 180.56 | 153.56 | 136.82 | — | 128.39 |
| | | WBB | 145.73 | 141.35 | 133.66 | — | 132.23 |
| | B | BLASSO | 26.35 | 39.23 | 50.77 | — | 73.63 |
| | | WBB | 21.80 | 34.82 | 47.76 | — | 75.29 |

structure and the sampling performance of approximate posterior summaries. It may provide a useful basis for further methodological development. Serial computations were used in the simulation above, and WBB was slightly slower than BLASSO (by a factor of 1.5 on CPU time); the computational advantage of WBB shows up in parallel computations, since weight vectors induce separate optimizations.

### 3.2. Diabetes Data

To further illustrate the WBB methodology, we apply it to the often-analyzed diabetes dataset (Efron et al., 2004). Data from $n = 442$ diabetes patients are available, with response a measure of

TABLE 3: 95% credible interval coverage.

| | | $p$ | 40 | 60 | 80 | 100 | 120 |
|---|---|---|---|---|---|---|---|
| $n = 50$ | A(i) | BLASSO | 0.91 | 0.92 | 0.93 | 0.94 | 0.94 |
| | | WBB | 0.92 | 0.92 | 0.93 | 0.94 | 0.95 |
| | A(ii) | BLASSO | 0.91 | 0.93 | 0.95 | 0.96 | 0.96 |
| | | WBB | 0.91 | 0.92 | 0.94 | 0.94 | 0.95 |
| | B | BLASSO | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | | WBB | 0.95 | 0.96 | 0.94 | 0.93 | 0.91 |
| $n = p/2$ | A(i) | BLASSO | 0.93 | 0.93 | 0.94 | — | 0.94 |
| | | WBB | 0.91 | 0.92 | 0.93 | — | 0.94 |
| | A(ii) | BLASSO | 0.92 | 0.94 | 0.95 | — | 0.96 |
| | | WBB | 0.91 | 0.93 | 0.94 | — | 0.95 |
| | B | BLASSO | 1.00 | 1.00 | 1.00 | — | 1.00 |
| | | WBB | 0.94 | 0.93 | 0.93 | — | 0.92 |

disease progression and with 10 baseline variables ($p = 10$), including age, sex, body mass index, average blood pressure, and six blood serum measurements. We apply Algorithm 1 with $T = 1,000$. (R code is in the Supplementary Material.) For each weight vector $\mathbf{w}$, the WBB estimate $\beta^*_{\mathbf{w}}$ is calculated using Equation (9) via the regularization method in the package `glmnet`.

$$\beta^{*,\text{common}}_{\mathbf{w}} := \arg\min_{\beta} \sum_{i=1}^{n} w_i (y_i - x_i'\beta)^2 + \lambda w_0 \sum_{j=1}^{p} |\beta_j| \,. \tag{9}$$

The regularization factor $\lambda$ is chosen by cross-validation with unweighted likelihood. The WBB is also performed with separate weights on each $|\beta_j|$,

$$\beta^{*,\text{separate}}_{\mathbf{w}} := \arg\min_{\beta} \sum_{i=1}^{n} w_i (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} w_{0,j} |\beta_j| \,.$$

As in the simulation study, WBB is compared to the Bayesian LASSO.

Figure 3 shows the results of all these three methods (the WBB with common/separate weight on prior terms, and Bayesian LASSO). Marginal posteriors for $\beta_j$'s are presented. For some coefficients there is very good agreement among the methods (e.g., `bmi` and `map`). One notable feature is that the WBB tends to introduce less sparsity than Bayesian LASSO. For example, the Bayesian LASSO posteriors of `age`, `tc`, `ldl`, `tch` and `glu` have higher spikes near zero compared with the WBB.

### 3.3. Trend Filtering

The generalized LASSO solves the optimization problem:

$$\begin{aligned} \beta^* &= \arg\min_{\beta} \{ l(y|\beta) + \lambda \phi(\beta) \} \\ &= \arg\min_{\beta} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|D\beta\|_1 \end{aligned}$$

 DOI: 10.1002/cjs.11570

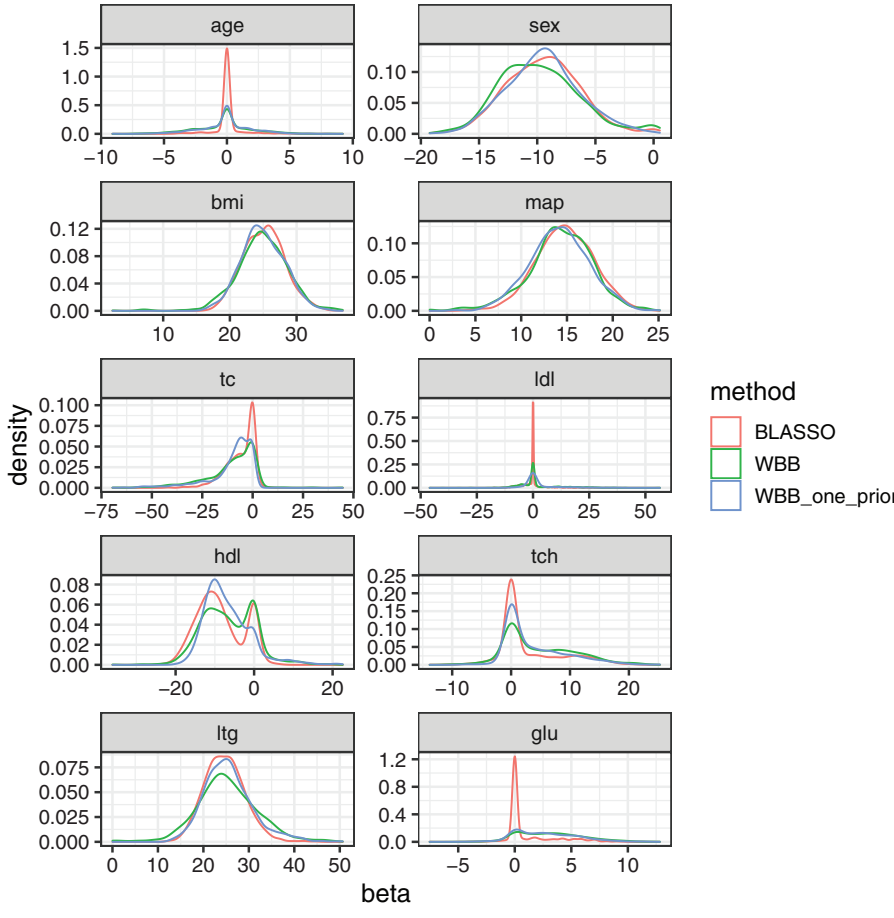


FIGURE 3: Diabetes example: the WBB (with common prior weight (blue) and separate prior weights (green)) and Bayesian LASSO (red) are used to draw from the marginal posteriors for $\beta_j$'s, j = 1,2,…, 10.

where $l(y|\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ is the negative log-likelihood. $D \in \mathcal{R}^{m\times p}$ is a penalty matrix and $\lambda\phi(\beta) = \lambda\|D\beta\|_1$ is the negative log-prior or regularization penalty. There are fast path algorithms for solving this problem (see `genlasso` package).

As a subproblem, polynomial trend filtering (Tibshirani, 2014; Polson & Scott, 2015) allows for piece-wise polynomial curve-fitting, where the knots and the parameters are chosen adaptively. Intuitively, the trend-filtering estimator is similar to an adaptive spline model: it penalizes the discrete derivative of order $k$, resulting in piecewise polynomials of higher degree for larger $k$.

Specifically, $X = I_p$ in the trend filtering setting and the data $y = (y_1, \ldots, y_p)$ are assumed to be meaningfully ordered from 1 to $p$. The penalty matrix is specially designed by the discrete $(k + 1)$th order derivative,

$$D^{(1)} = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 & 0 \\ 0 & -1 & 1 & \ldots & 0 & 0 \\ & & \ldots\ldots\ldots & & & \\ 0 & 0 & 0 & \ldots & -1 & 1 \end{bmatrix}_{(p-1)\times p}$$

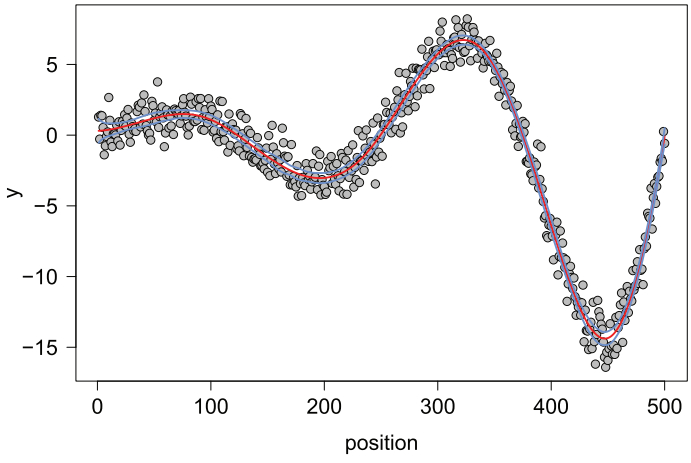


FIGURE 4: Cubic trend filtering: the red line is $\hat{\beta}_i$ for $i = 1, 2, \ldots, 500$; the blue line is $\hat{\beta}_i \pm 2 \times se$ where the standard errors are computed by WBB; $\lambda = 1,000$.

and $D^{(k+1)} = D^{(1)}D^{(k)}$ for $k = 1, 2, 3, \ldots$. For example, the log-prior in linear trend filtering is explicitly written as $\lambda \sum_{i=1}^{p-2} |\beta_{i+2} - 2\beta_{i+1} + \beta_i|$. For a general order $k > 1$,

$$\|D^{(k+1)}\beta\|_1 = \sum_{i=1}^{p-k-1} \Big| \sum_{j=i}^{i+k+1} (-1)^{(j-i)} \binom{k+1}{j-i} \beta_j \Big|.$$

WBB solves the following generalized LASSO problem in each draw:

$$\begin{aligned} \beta^*_{\mathbf{w}} &= \arg\min_{\beta} \frac{1}{2} \sum_{i=1}^{p} w_i (y_i - \beta_i)^2 + \lambda w_0 \|D^{(k)}\beta\|_1 \\ &= \arg\min_{\beta} \frac{1}{2} \|Wy - W\beta\|_2^2 + \lambda \|D^{(k)}\beta\|_1 \\ &= W^{-1} \arg\min_{\tilde{\beta}} \frac{1}{2} \|\tilde{y}_{\mathbf{w}} - \tilde{\beta}_{\mathbf{w}}\|_2^2 + \lambda \|\tilde{D}^{(k)}_{\mathbf{w}} \tilde{\beta}_{\mathbf{w}}\|_1 \end{aligned}$$

where $W = \text{diag}\left(\sqrt{w_1}, \ldots, \sqrt{w_p}\right) / \sqrt{w_0}$ and

$$\tilde{y}_{\mathbf{w}} = Wy,\ \tilde{\beta}_{\mathbf{w}} = W\beta,\ \tilde{D}^{(k)}_{\mathbf{w}} = D^{(k)}W^{-1}.$$

To illustrate, we simulate data $y_i$ from a Fourier series regression

$$y_i = \sin\left(\frac{4\pi}{500} i\right) \exp\left(\frac{3}{500} i\right) + \epsilon_i$$

for $i = 1, 2, \ldots, n = 500$, where $\epsilon_i \sim N(0, 2^2)$ are i.i.d. Gaussian deviates. The cubic trend filtering result is given in Figure 4. For each $i$, the WBB gives a group of estimates $\left\{\beta^{*,t}_{\mathbf{w}}(i)\right\}_{t=1}^{T}$ where $T$ is the total number of draws. The standard deviation of these weighted solutions constitutes a posterior standard deviation, or essentially a standard error for the estimator $\hat{\beta}_i$.

### 3.4. Deep Learning: MNIST Example

Deep learning is a form of machine learning that uses hierarchical abstract layers of latent variables to perform pattern matching and prediction. A Bayesian probabilistic perspective provides a number of insights into more efficient algorithms for optimization and hyper-parameter tuning. The general goal is to find a predictor of an output $y$ given a high dimensional input $x$. For a classification problem, $y \in \{1, 2, \ldots, K\}$ is a discrete variable and can be coded as a $K$-dimensional 0-1 vector. The model is as follows. Let $z^{(l)}$ denote the $l$th layer, and so $x = z^{(0)}$. The final output is the response $y$, which can be numeric or categorical. A deep prediction rule (Polson & Sokolov, 2017) is then

$$\begin{aligned} z^{(1)} &= f^{(1)}\Big(W^{(0)}x + b^{(0)}\Big), \\ z^{(2)} &= f^{(2)}\Big(W^{(1)}z^{(1)} + b^{(1)}\Big), \\ &\cdots \\ z^{(L)} &= f^{(L)}\Big(W^{(L-1)}z^{(L-1)} + b^{(L-1)}\Big), \\ \hat{y}(x) &= z^{(L)}. \end{aligned}$$

Here, $W^{(l)}$ are weight matrices, and $b^{(l)}$ are threshold or activation levels. $f^{(l)}$ is the activation function. Probabilistically, the output $y$ in a classification problem is generated by a probability model

$$p(y|x, W, b) \propto \exp\{-l(y|x, W, b)\}$$

where $l(y|x, W, b) = \sum_{i=1}^{n} l_i(y_i|x_i, W, b)$ is the negative cross-entropy,

$$l_i(y_i|x_i, W, b) = l_i(y_i, \hat{y}(x_i)) = \sum_{k=1}^{K} y_{ik} \log \hat{y}_k(x_i),$$

where $y_{ik}$ is 0 or 1. Adding the negative log-prior $\lambda\phi(W, b)$, the objective function (negative log-posterior) to be minimized by SGD is

$$\mathcal{L}_\lambda(y, \hat{y}) = \sum_{i=1}^{n} l_i(y_i, \hat{y}(x_i)) + \lambda\phi(W, b).$$

Accordingly, with each draw of weights $\mathbf{w}$, WBB provides the estimates $(W^*_{\mathbf{w}}, b^*_{\mathbf{w}})$ by solving the following optimization problem:

$$(W^*_{\mathbf{w}}, b^*_{\mathbf{w}}) = \arg\min_{W,b} \sum_{i=1}^{n} w_i l_i(y_i|x_i, W, b) + \lambda w_0 \phi(W, b).$$

We take the classic MNIST example (LeCun & Cortes, 2010) to illustrate the application of WBB in deep learning. The MNIST database of handwritten digits, available from Yann LeCun's website, has 60,000 training examples and 10,000 test examples. Here the high-dimensional $x$ is a normalized and centred fixed-size ($28 \times 28$) image and the output $\hat{y}$ is a 10-dimensional vector, where $i$th coordinate corresponds to the probability of that image being the $i$th digit.

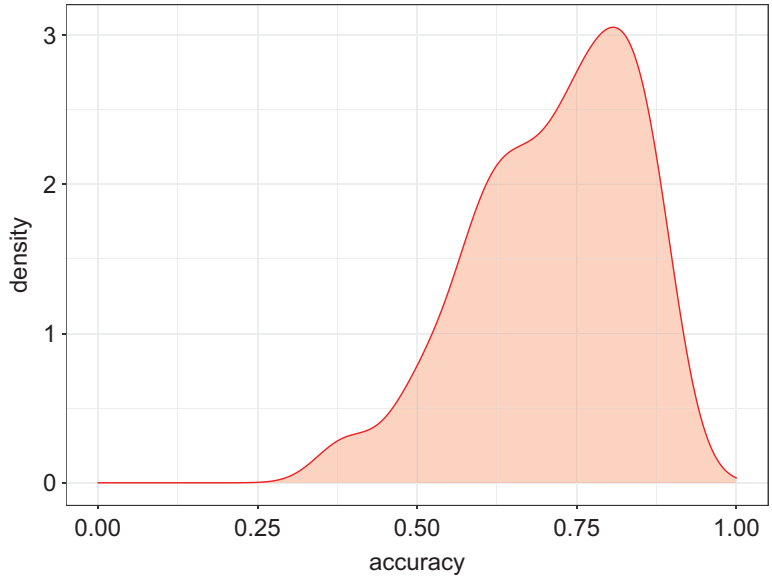


FIGURE 5: Posterior distribution of the classification accuracy by WBB. $n = 500$, $\lambda = 10^{-4}$.

For simplicity, we build a 2-layer neural network with layer sizes 128 and 64 respectively. Therefore, the dimensions of parameters are

$$W^{(0)} \in \mathcal{R}^{128\times 784},\ b^{(0)} \in \mathcal{R}^{128},$$
$$W^{(1)} \in \mathcal{R}^{64\times 128},\ b^{(1)} \in \mathcal{R}^{64},$$
$$W^{(2)} \in \mathcal{R}^{10\times 64},\ b^{(0)} \in \mathcal{R}^{10}.$$

The activation function $f^{(i)}$ is ReLU, $f(x) = \max\{0, x\}$, and the negative log-prior is specified as

$$\lambda\phi(W, b) = \lambda \sum_{l=0}^{2} \|W^{(l)}\|_2^2,$$

where we manually set $\lambda = 10^{-4}$.

Figure 5 shows the posterior distribution of the classification accuracy in the test dataset. We see that the test accuracies are centred around 0.75 and the posterior distribution is left-skewed. Furthermore, the accuracy is higher than 0.35 in 99% of the cases. The 95% interval is [0.407, 0.893]. Due to the simple 2-layer neural network structure, the classification accuracy is admittedly low compared with the state-of-the-art (e.g., Liang & Hu, 2015). This illustrative example shows how WBB can be easily implemented in deep learning. Implementing variational Bayes in deep learning is discussed in Polson & Sokolov (2017).

## 4. DISCUSSION

WBB provides a computationally attractive solution to scalable Bayesian inference (Minsker et al., 2014; Welling & Teh, 2011) whilst accounting for parameter uncertainty by maximizing a weighted posterior distribution. WBB can also be used in conjunction with proximal methods (Parikh & Boyd, 2014; Polson, Scott & Willard, 2015) to provide sparsity in high dimensional

 DOI: 10.1002/cjs.11570

statistical problems. With a similar ease of computation, WBB provides an alternative to ABC methods (Beaumont, 2009) and variational Bayes (VB) methods (Blei, Kucukelbir & McAuliffe, 2017). A fruitful area for future research is the comparison of WBB to recent extensions of the WLB for generalized Bayesian inference (Lyddon, Holmes & Walker, 2019).

For a wide range of non-smooth objective functions/statistical models, recent regularization methods provide fast, scalable algorithms for calculating estimates of the form Equation (1), which can also be viewed as posterior modes. Therefore as $\lambda$ varies we obtain a full regularization path as a form of prior sensitivity analysis. Furthermore, Strawderman, Wells & Schifano (2013) and Polson & Scott (2015) considered scenarios where posterior modes can be used as posterior means from augmented probability models. There may be useful interpretations of the random weights from the data-augmentation perspective.

Extending WBB asymptotics presents an exciting research opportunity. The argument in Section 2.5 relies on smoothness in both the sampling model and prior, and it retains fixed parameter dimension as $n$ increases. Theoretical guarantees remain unavailable for relatively large parameter dimension or for non-smooth penalty functions. Fortunately, groundwork has been done, for example by van der Pas, Kleijn & van der Vaart (2014), Narisetty & He (2014), and others on the asymptotic behaviour of the posterior distribution, and by Knight & Fu (2000) and others on sampling theory of optimization-based estimators,

## BIBLIOGRAPHY


Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., et al. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org.

Arnold, T. B. & Tibshirani, R. J. (2014). *genlasso: Path algorithm for generalized lasso problems*, R package version 1.3.

Barbe, P. & Bertail, P. (2012). *The Weighted Bootstrap*, Vol. 98. Springer Science & Business Media, Berlin.

Beaumont, M. A., Cornuet, J. M., Marin, J. M., & Robert, C. P. (2009). Adaptive approximate Bayesian computation. *Biometrika*, 96, 983–990.

Bertsekas, D. P., Nedi, A., & Ozdaglar, A. E. (2003). *Convex Analysis and Optimization*. Athena Scientific.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112, 859–877.

Boyd, S. & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge.

Carlin, B. P. & Polson, N. G. (1991). Inference for nonconjugate Bayesian models using the Gibbs sampler. *Canadian Journal of Statistics*, 19, 399–405.

Chollet, F. (2015). *Keras*, https://keras.io.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. J. (2004). *Diabetes dataset*, Available from R package *lars*.

Friedman, J., Hastie, T., & Tibshirani, R. J. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33, 1–22.

Gelman, A. & Meng, X. L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 163–185.

Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, Vol. 140, IET Digital Library, 107–113.

Green, P. J., Łatuszyński, K., Pereyra, M., & Robert, C. P. (2015). Bayesian computation: A summary of the current state, and samples backwards and forwards. *Statistics and Computing*, 25, 835–862.

Gribonval, R. & Machart, P. (2013). Reconciling "priors" & "priors" without prejudice?. *Advances in Neural Information Processing Systems*, 2193–2201.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, Boca Raton, FL.

Johnstone, I. M. (2010). High dimensional Bernstein–von Mises: Simple examples. *Institute of Mathematical Statistics Collections*, 6, 87.

Kleijn, B. J. K. & van der Vaart, A. W. (2012). The Bernstien–von-Mises theorem under misspecification. *Electronic Journal of Statistics*, 6, 354–381.

Knight, K. & Fu, W. (2000). Asymptotics for lasso-type estimators. *The Annals of Statistics*, 28, 1356–1378.

Lange, K. (2016). *MM Optimization Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436.

LeCun, Y. & Cortes, C. (2010). *MNIST handwritten digit database*, http://yann.lecun.com/exdb/mnist/.

Liang, M. & Hu, X. (2015). Recurrent convolutional neural network for object recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, New York, 3367–3375.

Lopes, H. F., Polson, N. G., & Carvalho, C. M. (2012). Bayesian statistics with a smile: A resampling-sampling perspective. *Brazilian Journal of Probability and Statistics*, 26, 358–371.

Lyddon, S., Holmes, C., & Walker, S. (2019). Generalized Bayesian updating and the loss likelihood bootstrap. *Biometrika*, 106, 465–478.

Minsker, S., Srivastava, S., Lin, L., & Dunson, D. (2014). Scalable and robust Bayesian inference via the median posterior. *International Conference on Machine Learning*, 1656–1664.

Mitchell, A. F. (1994). A note on posterior moments for a normal mean with double-exponential prior. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56, 605–610.

Muliere, P. & Secchi, P. (1996). Bayesian nonparametric predictive inference and bootstrap techniques. *Annals of the Institute of Statistical Mathematics*, 48, 663–673.

Narisetty, N. N. & He, X. (2014). Bayesian variable selection with shrinkage and diffusing priors. *The Annals of Statistics*, 42, 789–817.

Newton, M. A. & Raftery, A. E. (1994). Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, 3–48.

Nocedal, J. & Wright, S. J. (2006). Numerical optimization. *Springer Series in Operations Research*, 2nd ed., Springer, Berlin.

Parikh, N. & Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, 1, 127–239.

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

Polson, N. G. & Scott, J. G. (2015). Mixtures, envelopes and hierarchical duality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78, 701–727.

Polson, N. G., Scott, J. G., & Willard, B. T. (2015). Proximal algorithms in statistics and machine learning. *Statistical Science*, 30, 559–581.

Polson, N. G. & Sokolov, V. (2017). Deep learning: A Bayesian perspective. *Bayesian Analysis*, 12, 1275–1304.

Polson, N. G. & Sun, L. (2019). Bayesian $l_0$-regularized least squares. *Applied Stochastic Models in Business and Industry*, 35, 717–731.

Rubin, D. B. (1981). The Bayesian bootstrap. *The Annals of Statistics*, 9, 130–134.

Strawderman, R. L., Wells, M. T., & Schifano, E. D. (2013). Hierarchical Bayes, maximum a posteriori estimators, and minimax concave penalized likelihood estimation. *Electronic Journal of Statistics*, 7, 973–990.

Taddy, M., Chen, C. S., Yu, J., & Wyle, M. (2015). *Bayesian and empirical Bayesian forests*, arXiv preprint arXiv:1502.02312.

Tibshirani, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42, 285–323.

van der Pas, S., Kleijn, B., & van der Vaart, A. (2014). The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8, 2585–2618.

Wang, Y. & Swiler, L. (2018). Special issue on uncertainty quantification in multiscale system design and simulation. *ASCE–ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 4, 010301.

Welling, M. & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 681–688.

Wellner, J. & Zhang, T. (2012). Introduction to the special issue on sparsity and regularization methods. *Statistical Science*, 27, 447–449.

Wright, S. J. (2015). Coordinate descent algorithms. *Mathematical Programming*, 151, 3–34.

## APPENDIX

Here we consider how SGD may be deployed to minimize the penalized loss function, $\sum_{i=1}^{n} w_i l_i(y_i;\theta) + \lambda w_0 \phi(\theta)$. The method minimizes the function by taking a negative step along an estimate $g^k$ of the gradient $\nabla\left[\sum_{i=1}^{n} w_i l_i(y_i;\theta^k) + \lambda w_0 \phi(\theta^k)\right]$ at iteration $k$. The approximate gradient is estimated by calculating

$$g^k = \frac{n}{b_k} \sum_{i \in E_k} w_i \nabla l_i(y_i;\theta^k) + \lambda w_0 \frac{n}{b_k} \nabla \phi(\theta^k),$$

where $E_k \subset \{1,\dots,n\}$ and $b_k = |E_k|$ is the number of elements in $E_k$. When $b_k > 1$ the algorithm is called batch SGD and simply SGD otherwise. A usual strategy to choose subset $E$ is to go cyclically and pick consecutive elements of $\{1,\dots,T\}$, $E_{k+1} = [E_k \mod n] + 1$. The approximated direction $g^k$ is calculated using a chain rule (i.e., back-propagation) for deep learning. It is an unbiased estimator. Thus, at each iteration, the SGD updates the solution

$$\theta^{k+1} = \theta^k - t_k g^k.$$

For deep learning applications the step size $t_k$ (i.e., learning rate) is usually kept constant or some simple step size reduction strategy is used, $t_k = a\exp(-kt)$. Appropriate learning rates or the hyperparameters of the reduction schedule are usually found empirically from numerical experiments and observations of the loss function progression.

*Received 20 September 2018*
*Accepted 12 January 2020*