The Canadian Journal of Statistics

Vol. , 2019, Pages 1

La revue canadienne de statistique

Weighted Bayesian Bootstrap for Scalable Posterior Distributions

Blinded-A, Blinded-B and Blinded-C

Key words and phrases: Bayesian; Bootstrap; MCMC; Weighted Bootstrap; ABC; Trend Filtering; Deep Learning; TensorFlow; Regularization.

MSC 2010: Primary 62F40; secondary 6262F15

Abstract: We 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. We provide regularity conditions which apply to a range of machine learning and statistical models. We illustrate our methodology in regularized regression, trend filtering and deep learning. Finally, we conclude with directions for future research.

We would like to thank associated editor and referees for their detailed comments.

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. TensorFlow (TF) provides a numerical optimization framework for training and inference of models but does not provide uncertainty quantification. 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 here is to marry Bayesian uncertainty techniques with state-of-art optimization methods such as convex optimization and TensorFlow.

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 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 et al., 2015) that deploy stochastic gradient descent (SGD) and convex optimization methods for large-scale problems, such as on neural network architectures used in deep learning (LeCun *et al.* 2015). By linking random weighting with modern-day 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, 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 λ), which is usually difficult to compute in traditional Bayesian sensitivity analysis. We use predictive cross-validation techniques in this regard.

The rest of the paper 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). Finally, Section 4 concludes with directions for future research. Areas for future study include bootstrap filters in state-space models (Gordon *et al.* 1993) and comparison with the resampling-sampling perspective to sequential Bayesian inference (Lopes *et al.* 2012), etc.

2. WEIGHTED BAYESIAN BOOTSTRAP

2.1. Set up

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 $y = (y_1, y_2, \cdots, y_n)$ be an n-vector of outcomes and let $\theta = (\theta_1, \theta_2, ..., \theta_p)$ 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

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

where $l(y|\theta) = \sum_{i=1}^n l_i(y_i|\theta)$ is a measure of fit (or "empirical risk function") depending on θ 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 favorable bias-variance estimation tradeoff and provides extensive modeling flexibility (Wellner and Zhang, 2012). To accommodate contemporary applications we allow that $\phi(\theta)$ may have points in its domain where it fails to be differentiable (e.g., L^1 norm). If we treat the data, y, as arising from a probabilistic model parameterized by θ, then the likelihood function $p(y|\theta)$ yields the model-associated measure of fit $l(y|\theta) = -\log p(y|\theta)$. The maximimum likelihood estimator (MLE) is $\hat{\theta} := \mathrm{argmax}_\theta\; p(y|\theta)$, though of course this usually differs from the solution to (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

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

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 regession 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 θ^*, a solution to (1). See Gribonval and Machart (2013) for a nuanced view of the connection between (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 *et al.* 2003; Boyd & Vandenberghe 2004). For example if loss l is convex and differentiable in θ and penalty ϕ is convex, then a necessary and sufficient condition for θ^* to minimize $l(y|\theta) + \lambda\phi(\theta)$ is

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

where ∂ 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 θ^*, such as given by the normal equations in linear regression, (3) usefully guides algorithms that aim to solve θ^*. For example, under separability conditions on the penalty function, coordinate descent algorithms effectively solve for θ^*; see Wright (2015), or Hastie *et al.* (2015, chap. 5) for a statistical perspective. The optimization literature also characterizes θ^* as the fixed point of a proximal operator $\text{prox}_{\gamma\mathcal{L}}(\theta) = \arg\min_z\{\mathcal{L}(z) - \frac{1}{2\gamma}||z - \theta||_2^2\}$, which opens the door to powerful MM algorithms and related schemes; see Lange (2016, chap. 5), Polson & Scott (2015), Polson *et al.* (2015). Beyond convexity, the guarantees are weaker (e.g., local not global minima) and the algorithms are many (e.g., Nocedal and Wright, 2006). Gradient descent or stochastic gradient descent (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 *et al.* 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.* 2016).

2.4. WBB Algorithm

We now define the weighted Bayesian Bootstrap (WBB). Recalling the original objective function (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 $\mathbf{w} = (w_0, w_1, \cdots, w_n)$ 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 $p > n$ and $\phi(\theta)$ is separable as $\phi(\theta) = \sum_{j=1}^{p} \phi_j(\theta_j)$, we recommend individual weight on each prior regularization term $\phi_j(\theta_j)$, namely $\lambda \sum_{j=1}^{p} w_j^{\phi} \phi_j(\theta_j)$, to help with sparsity and to reduce the effect of occasionally large random weight w_0. 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 (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 (5), and we investigate the approximation numerically in a few examples in Section 3. Assuming this conjecture is true, we have immediately a straightforward, optimization-based algorithm for approximate posterior sampling:

Algorithm 1 Weighted Bayesian Bootstrap

Input:

data: $\mathcal{D} = (y, A)$

model structure: $\mathcal{M} = (\{l_i\}, \lambda, \phi)$

number of draws: T

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

Function $\mathbf{WBB}(\mathcal{D}, \mathcal{M}, T)$:

for all $t = 1$ to T **do**

Realize: $(u_0, u_1, \cdots, u_n) \sim_{i.i.d.}$ Uniform$(0, 1)$

Construct: $w_i \leftarrow \log(1/u_i), \forall i$. $\mathbf{w} = (w_0, w_1, \cdots, w_n)$

Compute: $\theta^{*,t} \leftarrow \arg\min \mathcal{L}_{\mathbf{w}}(\theta)$

end for

When optimization on the original problem (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 λ 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 (5) from an asymptotic perspective.

2.5. WBB Properties

Conditions under which the target posterior distribution (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 θ_n^*. Here $j(\theta)$ is the information per sample, and $y = (y_1, \cdots, y_n)$ denotes data. Johnstone (2010) studies Bernstein-von Mises theorem in high dimensional settings where p grows with n and the situation is very different. Centering on the posterior mode, rather than the MLE, improves accuracy in many cases (Bertail & Lo, 1991).

As to the WBB distribution, consider a one-term Taylor expansion of $\nabla\mathcal{L}_{\mathbf{w},n}(\theta)$ about the posterior mode θ_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 ∇ and ∇^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 (7), and leads to:

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

where $\tilde{R}_n$ is another error term. Following Newton and Raftery (1994), we recognize that the $\mathbf{w}$-induced variation in (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 (6), we have WBB variation:

$$\theta^*_{\mathbf{w},n}\Big|\, y \sim_{\text{approx}} N_p\left(\theta^*_n, J_n^{-1}(\theta^*_n)\right). \qquad (9)$$

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 and Raftery (1994), and differ only slightly in our use of the posterior mode θ^*_n in place of the maximum likelihood estimator $\hat{\theta}_n$, 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 (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 $(n+p)$-dimensional unit simplex, and thus correspond to a specific Dirichlet distribution. Exponential/Dirichlet weights are motivated from both inferential, related to the original Bayesian bootstrap (Rubin, 1981), and numerical, owing to benefits of smoothly varying weights. Other weight distributions may also be effective (Barbe and Bertail, 2012).

3. APPLICATIONS

We now illustrate our methodology with a number of scenarios to assess when WBB corresponds to a full Bayesian posterior distribution.

3.1. LASSO Experiment

First, 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:

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

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, WBB algorithm shrinks posterior means towards zero.

A simulation study is conducted in order to further investigate WBB with LASSO prior. We simulate data from a simple linear regression model,

$$Y = X\beta + \epsilon,\ \epsilon \sim N(0, \sigma_\epsilon^2 I)$$

where X is $N \times p$ and Y is $N \times 1$. The design matrix, $X = [X_1, X_2, ..., X_p]$, is generated from the distribution $X \sim N(0, \Sigma)$. The covariance between X_i and

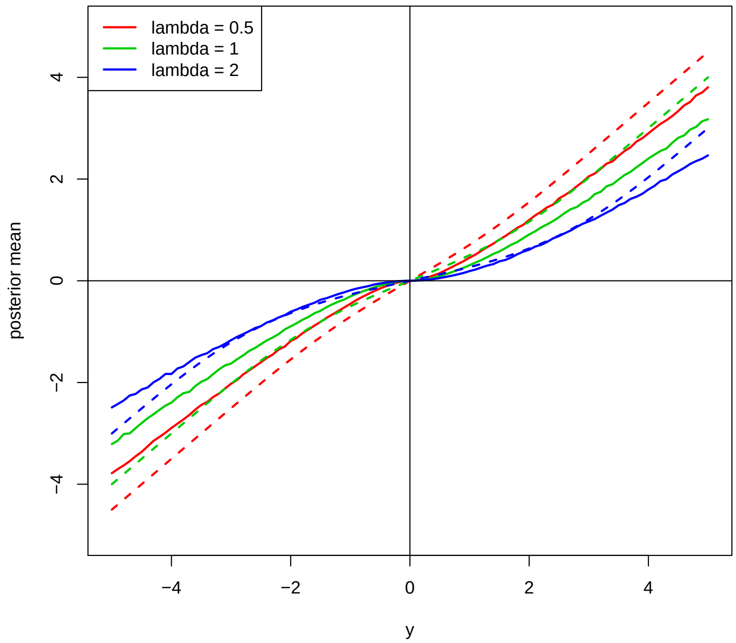

FIGURE 1: Normal means model with LASSO prior: WBB mean $E_{\mathbf{w}}(\theta^*_{\mathbf{w}}|y)$ (in solid lines) versus exact posterior mean $E(\theta|y)$ (in dashed lines).

X_j is set to be $\Sigma_{i,j} = 0.1 * 0.8^{|i-j|}$. Here we also assume i.i.d Gaussian noise where $\sigma_\epsilon = \sqrt{\|X\beta\|_2^2/(2N)}$. In this setting, the signal-noise ratio (SNR) is 2. In terms of the coefficient vector $\beta = [\beta_1, \beta_2, ..., \beta_p]'$, we consider both sparse and dense settings:

A(i). $\beta_j = 1$ for $1 \leq j \leq 10$ and $\beta_j = 0$ for $j > 10$.

A(ii). $\beta_j = 1$ for $1 \leq j \leq 5$, $\beta_j = 10$ for $6 \leq j \leq 10$ and $\beta_j = 0$ for $j > 10$.

B. $\beta_j = 1$ for $1 \leq j \leq p$.

For each setting of β, we set $p = 40, 60, 80, 100, 120$. To investigate the performance of our methodology under different sample size, we also choose two settings of N:

1. $N = 100$ for all p.
2. $N = p/2$ for all p.

For WBB with LASSO prior, we compare it with Bayesian LASSO (Park & Casella, 2008; Carlin & Polson, 1991). The two methods has different prior specifications,

$$\text{WBB} : \lambda\phi(\beta|\mathbf{w}) = \lambda \sum_{j=1}^{p} w_j^{\phi}|\beta_j|, \tag{10}$$

$$\text{BLASSO} : \phi(\beta|\sigma^2) = -\log\left(\frac{\lambda}{2\sigma}\right) + \frac{\lambda}{\sigma}\sum_{j=1}^{p}|\beta_j|. \tag{11}$$

Bayesian LASSO imposes a noninformative marginal prior on σ^2, $\pi(\sigma^2) = 1/\sigma^2$ and posterior distribution is sampled by a Gibbs sampler, with λ chosen by maximizing marginal likelihood. In WBB, λ is chosen via cross-validation, using standard unweighted LASSO. Here the original LASSO prior is separable. We give weights not only to each sample, but also to each individual β_j. In high-dimensional cases, an extremely large common w_0 multiplied to $\phi(\beta)$ can introduce extra sparsity to all marginal posteriors. We change the common prior weight to individual ones in hope of avoiding this issue. For comparison criteria, we present the estimation MSE of coefficients β (use posterior mean as our estimate), out-of-sample prediction MSE (test sets are of the same size as the corresponding training sets), and the (median, mean) of the 95% credible interval coverages. Fixing (p, n, β), the estimation procedure is repeated over B independent datasets $\{(X, Y)^{(b)}\}_{b=1}^{B}$. For each coordinate j, coverage is then calculated using $\{\beta_j^{(b)}\}_{b=1}^{B}$ and the two statistics are calculated across $1 \leq j \leq p$.

Results for estimation error, out-of-sample prediction error and credible interval coverage are displayed in Table (1), (2), (3), respectively. In all cases when $p > 40$, WBB has lower estimation error than BLASSO does. It also outperforms BLASSO in terms of out-of-sample prediction in general. The latter makes better predictions when $p \geq 100$ and the sparse β is not uniform across those non-zero coordinates. Credible intervals of both WBB and BLASSO don't have exact 95% coverages. WBB performs well when $p \geq 100$ and β is sparse, though it always gives narrow intervals. When $\beta_j = 1$ for all $1 \leq j \leq p$, BLASSO intervals are too wide while WBB intervals have accurate coverage at least when $n = 50$ and $40 \leq p \leq 60$.

3.2. Diabetes Data

To illustrate our methodology, we use weighted Bayesian Bootstrap (WBB) on the classic diabetes dataset (Efron et al., 2004). The measurements for 442 diabetes patients are obtained ($n = 442$), with 10 baseline variables ($p = 10$), such as age, sex, body mass index, average blood pressure, and six blood serum measurements.

TABLE 1: Coefficient Estimation Mean Squared Error (MSE)

	β	p	40	60	80	100	120
$N = 50$	A(i)	BLASSO	0.13	0.10	0.06	0.05	0.05
		WBB	0.18	0.06	0.05	0.04	0.03
	A(ii)	BLASSO	5.70	3.81	2.86	2.41	1.91
		WBB	7.02	3.01	2.21	1.84	1.57
	B	BLASSO	0.69	0.70	0.69	0.79	0.81
		WBB	1.41	0.50	0.52	0.54	0.55
$N = p/2$	A(i)	BLASSO	0.13	0.08	0.06	-	0.04
		WBB	0.13	0.07	0.05	-	0.03
	A(ii)	BLASSO	7.05	4.23	2.95	-	1.92
		WBB	6.15	3.86	2.51	-	1.52
	B	BLASSO	0.55	0.60	0.69	-	0.80
		WBB	0.66	0.60	0.57	-	0.51

TABLE 2: Out-of-Sample Prediction Mean Squared Error (MSE)

	β	p	40	60	80	100	120
$N = 50$	A(i)	BLASSO	3.21	3.29	3.63	3.71	3.84
		WBB	3.20	3.25	3.61	3.63	3.83
	A(ii)	BLASSO	120.22	129.77	129.21	130.59	133.64
		WBB	121.30	128.89	129.40	133.55	138.04
	B	BLASSO	19.98	33.12	47.18	63.50	80.49
		WBB	20.26	32.60	47.09	63.65	77.51
$N = p/2$	A(i)	BLASSO	4.57	3.92	4.05	-	3.47
		WBB	3.84	3.55	3.94	-	3.53
	A(ii)	BLASSO	184.01	151.70	142.93	-	128.91
		WBB	150.67	136.02	140.68	-	134.78
	B	BLASSO	27.93	37.57	51.24	-	76.36
		WBB	23.36	34.10	47.31	-	79.15

The likelihood function is given by

$$l(y|\beta) = \prod_{i=1}^{n} p(y_i|\beta)$$

TABLE 3: 95% Credible Interval Coverage

	β	p	40	60	80	100	120
$N = 50$	A(i)	BLASSO	0.91	0.93	0.93	0.94	0.95
		WBB	0.90	0.91	0.92	0.94	0.94
	A(ii)	BLASSO	0.91	0.93	0.95	0.96	0.96
		WBB	0.91	0.92	0.93	0.94	0.95
	B	BLASSO	1.00	1.00	1.00	1.00	1.00
		WBB	0.95	0.95	0.93	0.91	0.88
$N = p/2$	A(i)	BLASSO	0.94	0.94	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	0.90	0.92	0.93	-	0.95
	B	BLASSO	1.00	1.00	1.00	-	1.00
		WBB	0.90	0.91	0.91	-	0.90

where

$$p(y_i|\beta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(y_i - x_i'\beta)^2\right\}.$$

We draw 1000 sets of weights $\mathbf{w} = \{w_i\}_{i=1}^{n+1}$ where w_i's are i.i.d. exponentials. For each weight set, the weighted Bayesian estimate $\beta^*_{\mathbf{w}}$ is calculated using Equation (12) via the regularization method in the package `glmnet`.

$$\hat{\beta}_{\mathbf{w}} := \arg\min_{\beta} \sum_{i=1}^{n} w_i(y_i - x_i'\beta)^2 + \lambda w_0 \sum_{j=1}^{p} |\beta_j| \,. \tag{12}$$

The regularization factor λ is chosen by cross-validation with unweighted likelihood. The weighted Bayesian Bootstrap is also performed with individual weight on each $|\beta_j|$,

$$\hat{\beta}^{\phi}_{\mathbf{w}} := \arg\min_{\beta} \sum_{i=1}^{n} w_i(y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} w_j^{\phi}|\beta_j| \,. \tag{13}$$

As in the simulation study, WBB results are compared with Bayesian LASSO estimations.

Figure (2) shows the results of all these three methods (the weighted Bayesian Bootstrap with common / individual weight on prior terms , and Bayesian LASSO). Marginal posteriors for β_j's are presented. One notable feature is that the weighted Bayesian Bootstrap tends to introduce less sparsity than Bayesian LASSO does. For example, the Bayesian LASSO posteriors of `age, tc, ldl, tch` and `glu` have higher spikes located around 0, compared with the weighted Bayesian Bootstrap ones. For `tc` and `hdl`, multi-modes in the marginal posteriors are observed. In general, the posteriors with individual weight on prior terms are similar with those given common prior weights. This difference is naturally attributed to variation in the weight assigned to the log-prior penalty term.

3.3. Trend Filtering

The generalized LASSO solves the optimization problem:

$$\beta^* = \arg\min_{\beta} \; \{l(y|\beta) + \lambda\phi(\beta)\} \tag{14}$$

$$= \arg\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|D\beta\|_1 \tag{15}$$

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`

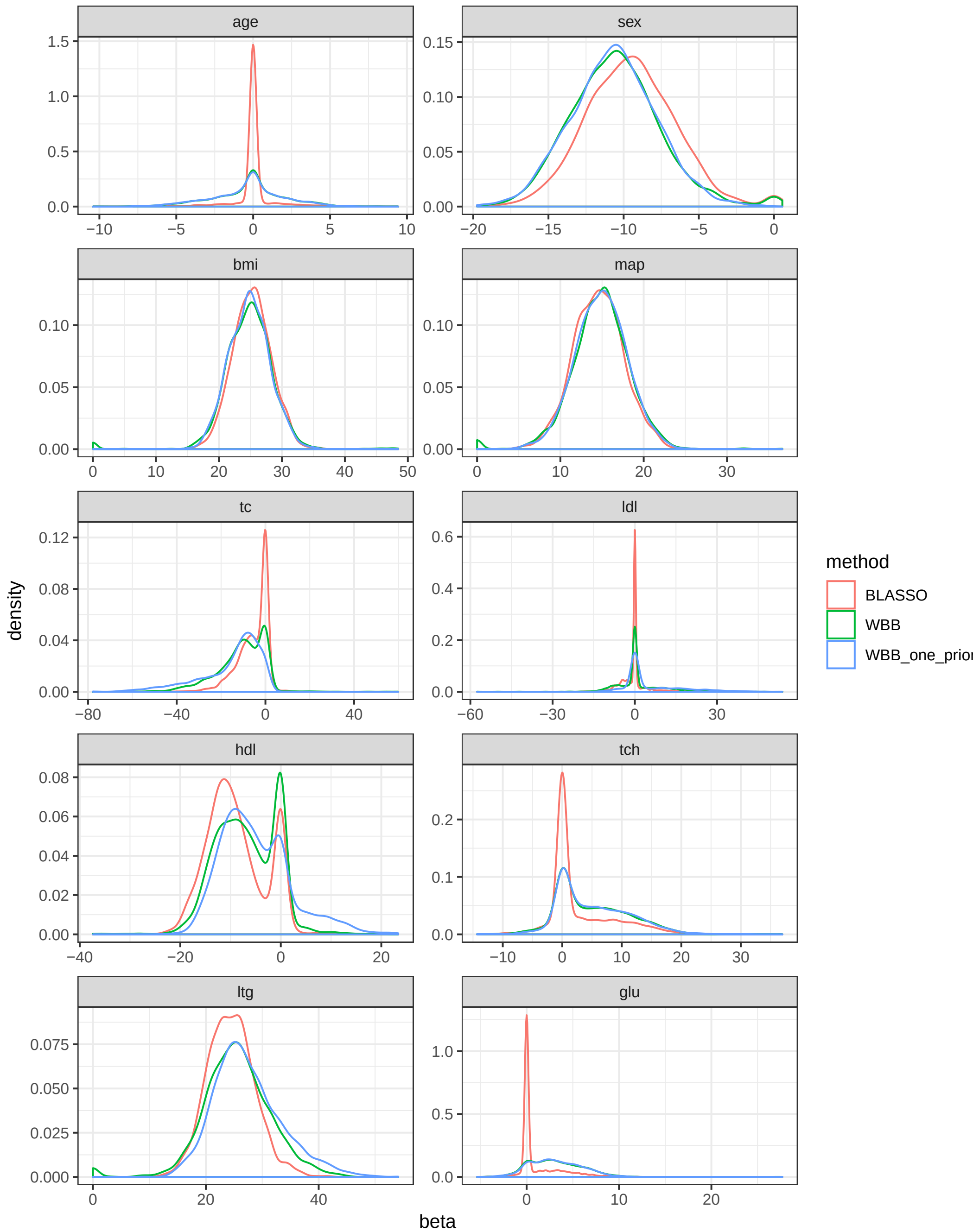

FIGURE 2: Diabetes example: the weighted Bayesian Bootstrap (with common prior weight and individual prior weights) and Bayesian LASSO are used to draw from the marginal posteriors for β_j's, j = 1,2,...10.

package).

As a subproblem, polynomial trend filtering (Tibshirani, 2014; Polson & Scott, 2015) is recently introduced 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.

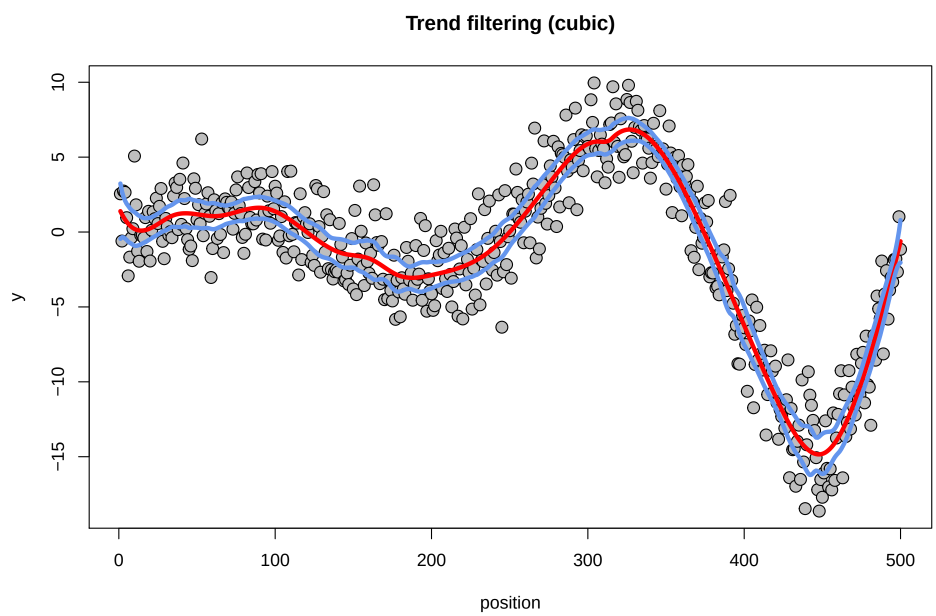

FIGURE 3: Cubic trend filtering: the red line is $\hat{\beta}_i$ for $i = 1, 2, ...500$; the blue line is $\hat{\beta}_i \pm 2 * se$ where the standard errors are computed by WBB. $\lambda = 1000$.

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

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

and $D^{(k+1)} = D^{(1)} D^{(k)}$ for $k = 1, 2, 3....$ 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 $W = \mathrm{diag}\left(\sqrt{w_1}/\sqrt{w_0}, \cdots, \sqrt{w_p}/\sqrt{w_0}\right)$ and

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

To illustrate our method, 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 $i = 1, 2, \cdots, n = 500$, where $\epsilon_i \sim N(0, 2^2)$ are i.i.d. Gaussian deviates. The cubic trend filtering result is given in Figure (3). For each i, the WBB gives a group of estimates $\{\beta^*_{\mathbf{w}}(i)\}_{j=1}^T$ 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, $y \in \{1, 2, ..., K\}$ 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

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

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 stochastic gradient descent 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 centered fixed-size (28×28) 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.

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 choose $\lambda = 10^{-4}$. Figure (4) shows the posterior distribution

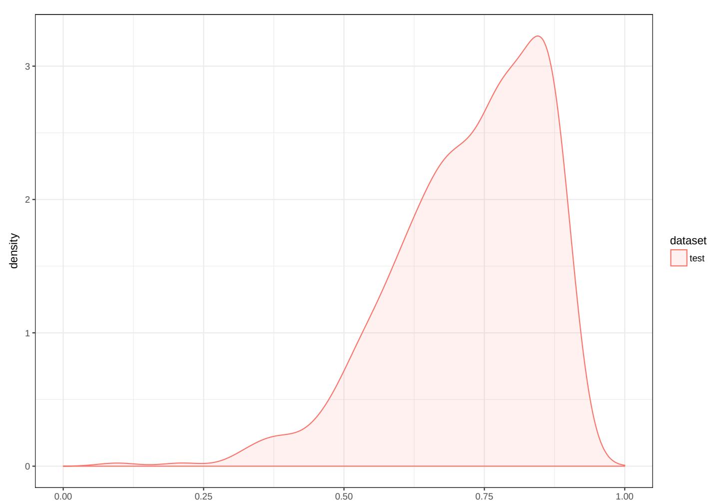

FIGURE 4: Posterior distribution of the classification accuracy. $n = 500, \lambda = 10^{-4}$.

of the classification accuracy in the test dataset. We see that the test accuracies are centered around 0.75 and the posterior distribution is left-skewed. Further-more, 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 those state-of-art ones (e.g. Lee *et al.* 2016; Liang & Hu, 2015). This illustration example however shows how WBB can be easily implemented in deep learning. Variational Bayes in deep learning is discussed in Polson & Sokolov (2017).

4. DISCUSSION

Weighted Bayesian Bootstrap (WBB) provides a computationally attractive solution to scalable Bayesian inference (Minsker *et al.* 2014; Welling & Teh, 2011) whilst accounting for parameter uncertainty by drawing samples from a weighted posterior distribution. WBB can also be used in conjunction with proximal methods (Parikh & Boyd, 2013; Polson *et al.* 2015) to provide sparsity in high dimensional statistical problems. With a similar ease of computation, WBB provides an alternative to ABC methods (Beaumont, 2009) and Variational Bayes (VB) methods (See Blei *et al* 2017 for a review of variational inference in Bayesian statistics). A fruitful area for future research is the comparison of approximate Bayesian computation with simulated Bayesian Bootstrap inference.

For a wide range of non-smooth objective functions/statistical models, recent regularization methods provide fast, scalable algorithms for calculating estimates of the form (1), which can also be viewed as the posterior mode. Therefore as λ varies we obtain a full regularization path as a form of prior sensitivity analysis.

Further, Strawderman *et al.* (2013) and Polson and 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 *et al.* (2014), Narisety & He (2014) and others on the asymptotic behaviour of the posterior distribution, and by Knight and 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., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Man?e, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Vi?egas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (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.

Beaumont, M. A., Cornuet, J.M., Marin,J.M. & Robert, C. P. (2009). Adaptive approximate bayesian computation. *Biometrika*, 96(4), 983–990.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518), 859-877.

Bertail, P. & Lo, A. Y. (1991). On Johnson's asymptotic expansion for a posterior distribution. *Centre de Recherche en Economie et Statistique*.

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

Boyd, S. & Vandenberghe, L. (2004) *Convex optimization*. Cambridge university press.

Carlin, B. P., & Polson, N. G. (1991). Inference for nonconjugate Bayesian models using the Gibbs sampler. *Canadian Journal of statistics*, 19(4), 399-405.

Chollet, François & others. (2015) Keras *https://keras.io*

Daniels, H. & Young, G. (1991). Saddlepoint approximation for the studentized mean, with an application to the bootstrap. *Biometrika*,78(1), 169–179.

Efron, B. (1981). Nonparametric standard errors and confidence intervals. *Canadian Journal of Statistics*, 9(2), 139–158.

Efron, B. (2012). Bayesian inference and the parametric bootstrap. *The Annals of Applied Statistics*, 6(4), 1971-1997.

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),1–22.

Fu, W.J., 1998. Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, 7(3), pp.397-416.

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. *In IEE Proceedings F (Radar and Signal Processing)*, 140(2) 107–113, IET Digital Library.

Gramacy, R. B. & Polson, N. G. (2012). Simulation-based regularized logistic regression. *Bayesian Analysis* , 7(3), 567–590.

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(4), 835-862.

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

Hastie, T., Tibshirani, R. & Wainwright, M., (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

Johnstone, I. M. (2010). High dimensional Bernstein-von Mises: simple examples. *Institute of Mathematical Statistics collections*, 6, 87.

Lange, K. (2016). *MM Optimization Algorithms*. Vol. 147, SIAM.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436.

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

Liang, M., & Hu, X. (2015). Recurrent convolutional neural network for object recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 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(4) 358–371.

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)*, 605–610.

Newton, M. A. & Raftery,, 1994 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*, 2nd edition. Springer series in Operations Research.

Parikh, N. & Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3), 127–239.

Park, T., & Casella, G. (2008). The Bayesian lasso. *Journal of the American Statistical Association*, 103(482), 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(4), 701–727.

Polson, N. G., Scott, J. G., & Willard, B. T. (2015). Proximal algorithms in statistics and machine learning. *Statistical Science*, 30(4), 559–581.

Polson, N. G., Scott, J. G., & Windle, J. (2014). The Bayesian bridge. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4), 713–733.

Polson, N. G. & Sokolov, V. (2017). Deep learning: A bayesian perspective. *Bayesian Analysis*,12(4), 1275–1304.

Rubin, D. B. (1981). The Bayesian bootstrap. *The Annals of Statistics*, 9(1), 130–134.

Strawderman, R. L., Wells, M. T., & Schifano, E. D. (2013). Hierarchical Bayes, maxi- mum 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(1), 285–323.

Tibshirani, R.J. (2017). Dykstra's Algorithm, ADMM, and Coordinate Descent: Connections, Insights, and Extensions. In *Advances in Neural Information Processing Systems*, pp. 517-528.

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(2), 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(1), 010301.

Welling, M. & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. *In 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(1):3-34.

APPENDIX

Stochastic gradient descent (SGD) method or its variation is typically used to find the deep learning model weights by minimizing the penalized loss function, $\sum_{i=1}^{n} w_i l_i(y_i;\theta) + \lambda w_p \phi(\theta)$. The method minimizes the function by taking a negative step along an estimate g^k of the gradient $\nabla\left[\sum_{i=1}^{n} w_i l_i(y_i;\theta^k) + \lambda w_p \phi(\theta^k)\right]$ at iteration k. The approximate gradient is estimated by calculating

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

Where $E_k \subset \{1,\ldots,n\}$ 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 $\{1,\ldots,T\}$, $E_{k+1} = [E_k \mod n] + 1$. The approximated direction g^k is calculated using a chain rule (aka 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 (a.k.a 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 reduction schedule are usually found empirically from numerical experiments and observations of the loss function progression.