Weighted Bayesian Bootstrap

Michael Newton
University of Wisconsin Madison[*]

Nicholas G. Polson
Booth School of Business University of Chicago

Jianeng Xu
Booth School of Business University of Chicago

First Draft: October 2016
This Draft: February 2018

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 such as TensorFlow. We provide regularity conditions which apply to a wide 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.

Keywords: Bayesian, Bootstrap, MCMC, Weighted Bootstrap, ABC, Trend Filtering, Deep Learning, TensorFlow, Regularization.

[*]Newton is Professor of Biostatistics and Medical Informatics at University of Wisconsin-Madison. email: newton@biostat.wisc.edu. Polson is Professor of Econometrics and Statistics at the Chicago Booth School of Business. email: ngp@chicagobooth.edu. Xu is at the Chicago Booth School of Business. email: jianeng@uchicago.edu.

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. Whilst there are computationally fast and scalable algorithms for training models in a wide variety of contexts, uncertainty assessment is still required. 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) in Tensor-Flow can also be used to provide uncertainty quantification.

Our work builds on Newton and Raftery (1991) who provide a weighted likelihood Bootstrap (WLB) method for Bayesian inference. They develop a weighted likelihood Bootstrap algorithm together with the appropriate asymptotic analysis to show that such an algorithm provides efficient posterior samples. Their bootstrap procedure exploits the fact that the posterior distribution centered at the maximum likelihood estimate (MLE) has a second order expansion that also depends on the prior and its derivative. The weighted Bayesian Bootstrap (WBB) calculates a series of posterior modes rather than MLEs. This has the advantage that high dimensional posterior modes are readily available particularly using the regularized estimates are fast to compute from convex optimization methods or stochastic gradient descent (SGD) for neural network architectures such as deep learning. By linking WLB and WBB, with modern-day optimization to calibrate estimate, we provide a framework for uncertainty quantification.

Uncertainty estimates are provided at little to no extra cost. Quantifying uncertainty is typically unavailable in a purely regularization optimization method. Another feature that is straightforward to add is a regularization path across hyper-parameters. This is so much easier than traditional Bayesian to do prior sensitivity analysis where hyper-parameters are hard to assess. Rather we use predictive cross-validation techniques.

The rest of the paper is outlined as follows. Section 2 develops our weighted Bayesian Bootstrap (WBB) algorithm. Section 3 provides an application to high dimensional sparse regression, trend filtering and deep learning. Finally, Section 4 concludes with directions for future research. One area for future study is Bootstrap filters in state-space models (Gordon et al. (1993))

2   Weighted Bayesian Bootstrap

Let y an n-vector of outcomes, θ denotes a d-dimensional parameter of interest and A a fixed $n \times d$ matrix 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 machine learning and

statistical problems can be expressed in the form

$$\underset{\theta \in \mathcal{R}^d}{\text{minimize}} \quad \sum_{i=1}^{n} \log f(y_i; a_i^\top \theta) + \lambda \phi(\theta), \tag{1}$$

where $l(\theta) = \sum_{i=1}^{n} \log f(y_i; a_i^\top \theta)$ is a measure of fit (or "empirical risk function") depending implicitly on A and y. The penalty function or regularization term, $\lambda \phi(\theta)$, effects a favorable bias-variance tradeoff. We allow for the possibility that $\phi(\theta)$ may have points in its domain where it fails to be differentiable.

Suppose that we observe data, $y = (y_1, \ldots, y_n)$ from a model parameterized by θ. For example, we might have a probabilistic model that depends on a parameter, θ, where $f(y|\theta)$ is known as the likelihood function. Equivalently, we can define a measure of fit $l(\theta) = \log f(y|\theta)$. We will make use of the following

(i) Let $\hat{\theta}_n := \text{argmax}_\theta f(y|\theta)$ be the MLE,

(ii) Let $\theta_n^* := \text{argmax}_\theta p(\theta|y)$ be the posterior mode,

(iii) Let $\bar{\theta}_n := E(\theta|y)$ be the posterior mean.

We now develop a key duality between regularization and posterior bootstrap simulation.

2.1 Bayesian Regularization Duality

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

$$f(y|\theta) = p(y|\theta) \propto \exp\{-l(y|\theta)\}, \quad p(\theta) \propto \exp\{-\lambda \phi(\theta)\}$$
$$p(\theta|y) \propto \exp\{-(l(y; \theta) + \lambda \phi(\theta))\}.$$

The prior is not necessarily proper but the posterior, $p(\theta|y) \propto p(y|\theta)p(\theta)$, may still be proper. This provides an equivalence between regularization and Bayesian methods. For example, regression with a least squares log-likelihood subject to a penalty such as an L^2-norm (ridge) Gaussian probability model or L^1-norm (lasso) double exponential probability model. We then have

$$\hat{\theta}_n = \underset{\theta \in \Theta}{\arg\min} \, l(\theta), \tag{2}$$
$$\theta_n^* = \underset{\theta \in \Theta}{\arg\min} \, \{l(\theta) + \lambda \phi(\theta)\}. \tag{3}$$

Let ∂ be the subdifferential operator. Then a necessary and sufficient condition for θ^* to minimize $l(\theta) + \lambda \phi(\theta)$ is

$$0 \in \partial \{l(\theta) + \lambda \phi(\theta)\} = \nabla l(\theta) + \lambda \partial \phi(\theta) \tag{4}$$

the sum of a point and a set. The optimization literature characterizes θ^* as the fixed point of a proximal operator $\theta^* = \text{prox}_{\gamma\phi}\{\theta^* - \lambda\nabla f(\theta^*)\}$, see Polson and Scott (2015) and Polson, Scott, and Willard (2015) for further discussion.

A general class of natural exponential family models can be expressed in terms of the Bregman divergence of the dual of the cumulant transform. Let ϕ be the conjugate Legendre transform of ψ. Hence $\psi(\theta) = \sup_\mu \left(\mu^\top\theta - \phi(\mu)\right)$. Then we can write

$$p_\psi(y|\theta) = \exp\left(y^\top\theta - \psi(\theta) - h_\psi(y)\right)$$
$$= \exp\left\{\inf_\mu \left((y-\mu)^\top\theta - \phi(\mu)\right) - h_\psi(y)\right\}$$
$$= \exp\left(-D_\phi(y, \mu(\theta)) - h_\phi(y)\right)$$

where the infimum is attained at $\mu(\theta) = \phi'(\theta)$ is the mean of the exponential family distribution. We rewrite $h_\psi(y)$ is terms of the correction term and $h_\phi(y)$. Here D_ϕ can be interpreted as a Bregman divergence.

For a wide range of non-smooth objective functions/statistical models, recent regularization methods provide fast, scalable algorithms for calculating estimates of the form (3), 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.

Strawderman et al. (2013) and Polson et al. (2015) considered scenarios where posterior modes can be used as posterior means from augmented probability models. Moreover, in their original foundation of the Weighted Likelihood Bootstrap (WLB), Newton and Raftery (1991) introduced the concept of the implicit prior. Clearly this is an avenue for future research.

2.2 WBB Algorithm

We now define the weighted Bayesian Bootstrap (WBB). Following Newton and Raftery (1991), we construct a randomly weighted posterior distribution denoted by

$$\mathbf{w} = (w_1, ..., w_n, w_p),\ p_\mathbf{w}(\theta|y) \propto \prod_{i=1}^n f(y_i|\theta)^{w_i} p(\theta)^{w_p}$$

where the weights $w_p, w_i \sim Exp(1)$ are randomly generated weights. We have used the fact that for i.i.d. observations, the likelihood can be factorized as $f(y|\theta) = \prod_{i=1}^n f(y_i|\theta)$. This is not crucial for our analysis but is a common assumption. Let $\theta_{w,n}^*$ denote the mode of this regularized distribution. Again, there is an equivalence

$$\theta_{w,n}^* := \arg\max_\theta p_w(\theta|y) \equiv \arg\min_\theta \sum_{i=1}^n w_i l_i(y_i; \theta) + \lambda w_p \phi(\theta)$$

4

where $l_i(\theta) = -\log f(y_i|\theta)$ and $\lambda\phi(\theta) = -\log p(\theta)$. Note that we have a weighted likelihood and a new regularization parameter, λw_p.

The crux of our procedure is to create a sample of the weighted posterior modes $\{\theta^*_{w,n}\}$ (computationally cheap as each sub-problem can be solved via optimization. Our main result is the following:

Algorithm: Weighted Bayesian Bootstrap (WBB)

1. Sample $\mathbf{w} \sim Multi_{n+1}(\mathbf{p})$ via exponentials. $w_i \sim Exp(1)$. Set $w_i := w_i/\bar{w}$.

2. Solve $\theta^*_{w,n} = \arg\min_\theta \sum_{i=1}^n w_i l_i(y_i; \theta) + \lambda w_p \phi(\theta)$.

The WBB algorithm is fast (and scalable) to compute a regularized estimator. For a large number of popular priors, the minimizing solution $\theta^*_{w,n}$ in the second step can be directly obtained via regularization packages such as `glmnet` by Trevor Hastie and `genlasso` by Taylor Arnold. When the likelihood function or the prior is specially designed, Stochastic Gradient Descent (SGD) is powerful and fast enough to solve the minimization problem. It can be easily implemented in TensorFlow once the objective function is specified. See Appendix (5) and Polson and Sokolov (2017) for further discussion.

The next section builds on Newton and Raftery (1991, 1994) and derives asymptotic properties of the weighted Bayesian Bootstrap. We simply add the regularized factor. To choose the amount of regularization λ, we can use the marginal likelihood $m_\lambda(y)$, estimated by bridge sampling (Gelman and Meng (1998)) or simply using predictive cross-validation.

2.3 WBB Properties

The following proposition (which follows from the Theorem 2 in Newton and Raftery (1994)) summaries the properties of WBB.

Proposition *The weighted Bayesian Bootstrap draws are approximate posterior samples*

$$\left\{\theta^{*(k)}_{w,n}\right\}_{k=1}^K \sim p(\theta|y).$$

Now we consider 'large n' properties. The variation in the posterior density $p(\theta|y) \propto e^{-nl_n(\theta)}p(\theta)$ for sufficiently large n will be dominated by the likelihood term. Expanding $l_n(\theta)$ around its maximum, $\hat{\theta}$, and defining $J_n(\hat{\theta}) = nj(\hat{\theta})$ as the observed information matrix gives the traditional normal approximation for the posterior distribution

$$\theta \sim N_d\left(\hat{\theta}_n, J_n^{-1}(\hat{\theta})\right)$$

where $\hat{\theta}_n$ is the MLE. A more accurate approximation is obtained by expanding around the posterior mode, θ^*, which we will exploit in our weighted Bayesian Bootstrap. Now we have the asymptotic distributional approximation

$$\theta \sim N_d\left(\theta^*, J_n^{-1}(\theta^*)\right)$$

where $\theta_n^* := \arg\max_\theta p(\theta|y)$ is the posterior mode.

The use of the posterior mode here is crucially important as it's the mode that is computationally available from TensorFlow and Keras. Approximate normality and second order approximation also holds, see Johnson (1970), Bertail and Lo (1991) and Newton and Raftery (1991, 1994) for future discussion. Specifically,

$$\sqrt{nI(\hat{\theta}_n)}\left(\theta_n^* - \hat{\theta}_n\right) \stackrel{D}{=} Z$$

where $Z \sim N(0,1)$ is a standard Normal variable. The conditional posterior satisfies $\mathbb{P}\left(|\theta_n^* - \hat{\theta}_n| > \epsilon\right) \to 0$ for each $\epsilon > 0$ as $n \to \infty$. In the 'large p' case, a number of results are available for posterior concentration, for example, see Van Der Pas et al. (2014) for sparse high dimensional models.

3 Applications

Consider now a number of scenarios to assess when WBB corresponds to a full Bayesian posterior distribution.

3.1 Lasso

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. standard exponential weights w_1 and w_2, the weighted posterior mode θ_w^* is

$$\theta_w^* = \arg\min_{\theta \in \Theta} \left\{\frac{w_1}{2}(y - \theta)^2 + \lambda w_2|\theta|\right\}.$$

This case is sufficiently simple to work out the exact WBB solution in terms of soft thresholding:

$$\theta_w^* = \begin{cases} y - \lambda w_2/w_1 & \text{if } y > \lambda w_2/w_1, \\ y + \lambda w_2/w_1 & \text{if } y < -\lambda w_2/w_1, \\ 0 & \text{if } |y| \leq \lambda w_2/w_1. \end{cases}$$

The WBB mean $E_w(\theta_w^*|y)$ is approximated by the sample mean of $\{\theta_w^{*(k)}\}_{k=1}^K$. 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 gives sparser posterior means.
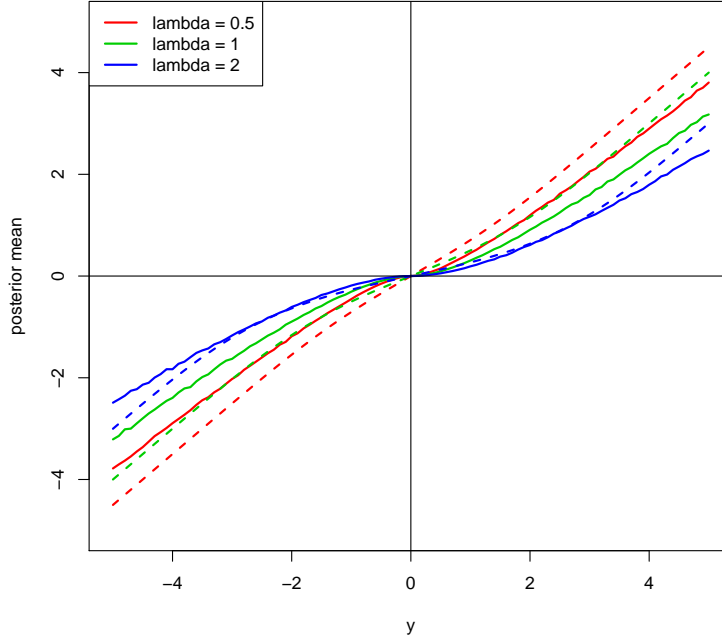

Figure 1: Normal means model with lasso prior: WBB mean $E_w(\theta_w^*|y)$ (in solid lines) versus exact posterior mean $E(\theta|y)$ (in dashed lines).

3.2 Diabetes Data

To illustrate our methodology, we use weighted Bayesian Bootstrap (WBB) on the classic diabetes dataset. 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.

The likelihood function is given by

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

where

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

We draw 1000 sets of weights $\mathbf{w} = \{\mathbf{w_i}\}_{\mathbf{i=1}}^{\mathbf{n+1}}$ where w_i's are i.i.d. exponentials. For each weight set, the weighted Bayesian estimate $\hat{\beta}_w$ is calculated using (5) via the regularization method in the package glmnet.

$$\hat{\beta}_{\mathbf{w}} := \arg\min_{\beta} \sum_{i=1}^{n} w_i(y_i - x_i'\beta)^2 + \lambda w_{n+1} \sum_{j=1}^{p} |\beta_j|. \tag{5}$$

The regularization factor λ is chosen by cross-validation with unweighted likelihood. The weighted Bayesian Bootstrap is also performed with fixed prior, namely, w_{n+1} is set to be 1 for all bootstrap samples. Polson et al. (2014) analyze the same dataset using the Bayesian Bridge estimator and suggest MCMC sampling from the posterior.

To compare our WBB results we also run the Bayesian bridge estimation. Here the Bayesian setting we use is

$$p(\beta, \sigma^2) = p(\beta|\sigma^2)p(\sigma^2), \text{ where } p(\sigma^2) \propto 1/\sigma^2.$$

The prior on β, with suitable normalization constant C_α, is given by

$$p(\beta) = C_\alpha \exp(-\sum_{j=1}^{p} |\beta_j/\tau|^\alpha).$$

The hyper-parameter is drawn as $\nu = \tau^{-\alpha} \sim \Gamma(2,2)$, where $\alpha = 1/2$.

Figure 3 shows the results of all these three methods (the weighted Bayesian Bootstrap with fixed prior / weighted prior and the Bayesian Bridge). Marginal posteriors for β_j's are presented. One notable feature is that the weighted Bayesian Bootstrap tends to introduce more sparsity than Bayesian Bridge does. For example, the weighted Bayesian Bootstrap posteriors of age, ldl and tch have higher spikes located around 0, compared with the Bayesian Bridge ones. For tc, hdl, tch and glu, multi-modes in the marginal posteriors are observed. In general, the posteriors with fixed priors are more concentrated than those with randomly weighted priors. This difference is naturally attributed to the certainty in the prior weights.

3.3 Trend Filtering

The generalized lasso solves the optimization problem:

$$\hat{\beta} = \underset{\beta}{\arg\min} \; \{l(y|\beta) + \lambda\phi(\beta)\} \tag{6}$$

$$= \underset{\beta}{\arg\min} \; \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|D\beta\|_1 \tag{7}$$

where $l(y|\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ is the log-likelihood. $D \in \mathbb{R}^{m \times p}$ is a penalty matrix and $\lambda\phi(\beta) = \lambda\|D\beta\|_1$ is the 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 and 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.

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,

$$D^{(1)} = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \multicolumn{6}{c}{\dotfill} \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}_{(p-1)\times p}$$

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:

$$\beta_w^* = \underset{\beta}{\arg\min} \; \frac{1}{2} \sum_{i=1}^{p} w_i (y_i - \beta_i)^2 + \lambda w_{p+1} \|D^{(k)}\beta\|_1$$

$$= \underset{\beta}{\arg\min} \; \frac{1}{2}\|Wy - W\beta\|_2^2 + \lambda\|D^{(k)}\beta\|_1$$

$$= W^{-1} \times \underset{\tilde{\beta}}{\arg\min} \; \frac{1}{2}\|\tilde{y}_w - \tilde{\beta}_w\|_2^2 + \lambda\|\tilde{D}_w^{(k)}\tilde{\beta}_w\|_1$$

where

$$W = diag(\sqrt{w_i}/\sqrt{w_{p+1}}, ...\sqrt{w_p}/\sqrt{w_{p+1}})$$

and

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

Now we simulate the data y_i as

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

for $i = 1, 2, ...500$, where $\epsilon_i \sim N(0, 2^2)$ are i.i.d. Gaussian noises. The cubic trend filtering result is given in Figure (4).

For each i, the weighted Bayesian Bootstrap gives a group of estimates $\{\beta_w^*(i)\}_{j=1}^T$ where T is the total number of draws. The standard errorof $\hat{\beta}_i$ is easily computed using these weighted bootstrap estimates.

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. Polson and Sokolov (2017) take a Bayesian probabilistic perspective and provide a number of insights into more efficient algorithms for optimization and hyper-parameter tuning.

The general goal is to finds 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. 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 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. Probabilistically, the output Y in a classification problem is generated by a probability model

$$p(Y|Y^{W,b}(X)) \propto \exp\{-l(Y|X, W, b)\}$$

where $l(Y|X, W, b) = \sum_{i=1}^{n} l_i(Y_i|X_i, W, b)$ is the negative cross-entropy,

$$l_i(Y_i|X_i, W, b) = l_i(Y_i, \hat{Y}(X_i)) = Y_i \log \hat{Y}(X_i)$$

Adding the negative log-prior $\lambda\phi(W, b)$, the objective function to be minimized by stochastic gradient descent is

$$\mathcal{L}_\lambda(Y, \hat{Y}) = \sum_{i=1}^{n} Y_i \log \hat{Y}(X_i) + \lambda\phi(W, b).$$

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

$$(\hat{W}_{\mathbf{w}}, \hat{b}_{\mathbf{w}}) = \arg\min_{W,b} \sum_{i=1}^{n} w_i Y_i \log \hat{Y}(X_i) + \lambda w_p \phi(W, b)$$

We take the classic MNIST example to illustrate the application of WBB in deep learning. The MNIST database of handwritten digits, available from Yann LeCun's website, has a training set of 60,000 examples, and a test set of 10,000 examples. Here the high-dimensional X is a normalized and centered fixed-size (28) image and the output 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

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

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 $\lambda = 10^{-4}$.

Figure (5) shows the posterior distribution 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. Furthermore, the accuracy is higher than 0.35 in 99% of the cases. The 95% interval is [0.407, 0.893].

11

4 Discussion

Our weighted Bayesian Bootstrap accounts for parameter uncertainty in machine learning and statistics problems by approximating samples from a posterior distribution. This can be computationally expensive in contrast to the optimization methods based on convex optimization (Parikh and Boyd (2013), Polson et al. (2015)). With a similar ease of computation, our method provides an alternative to ABC methods (Beaumont et al. (2009)) for Bayesian inference and a fruitful area for future research is the comparison with these methods.

References

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

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

Daniels, H. and G. Young (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.

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

Gordon, N. J., D. J. Salmond, and A. F. Smith (1993). Novel approach to nonlinear/nongaussian Bayesian state estimation. In *IEE Proceedings F-Radar and Signal Processing*, Volume 140, pp. 107–113. IET.

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

Hans, C. (2009). Bayesian lasso regression. *Biometrika 96*(4), 835–845.

Johnson, R. A. (1970). Asymptotic expansions associated with posterior distributions. *The annals of mathematical statistics 41*(3), 851–864.

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. (1991). *The weighted likelihood bootstrap and an algorithm for prepivoting*. Ph. D. thesis, Department of Statistics, University of Washington, Seattle.

Newton, M. A. and A. E. Raftery (1991). Approximate Bayesian inference via the weighted likelihood bootstrap. *Technical Report 199*.

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

Parikh, N. and S. Boyd (2013). Proximal algorithms, in foundations and trends in optimization.

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

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

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

Polson, N. G. and V. Sokolov (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., M. T. Wells, and E. D. Schifano (2013). Hierarchical bayes, maximum a posteriori estimators, and minimax concave penalized likelihood estimation. *Electronic Journal of Statistics 7*, 973–990.

Tibshirani, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics 42*(1), 285–323.

Van Der Pas, S., B. Kleijn, and A. Van Der Vaart (2014). The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics 8*(2), 2585–2618.

West, M. (1992). Modelling with Mixtures. *In Bayesian Statistics 4*, 503–524.

5 Appendix: Stochastic Gradient Descent (SGD)

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.
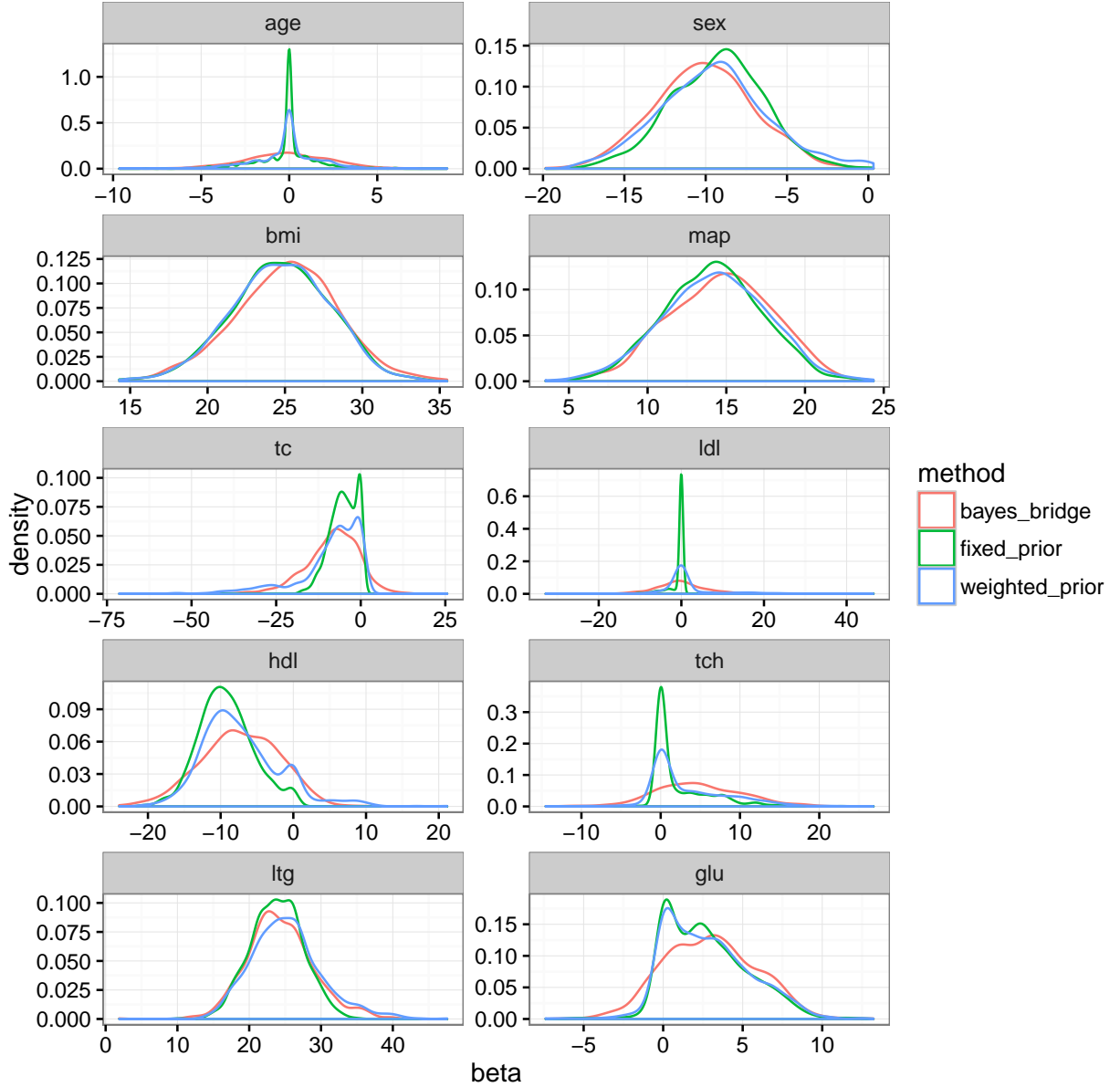
Figure 2: Diabetes example: the weighted Bayesian Bootstrap (with fixed prior and weighted prior) and Bayesian Bridge are used to draw from the marginal posteriors for β_j's, j = 1,2,...10.
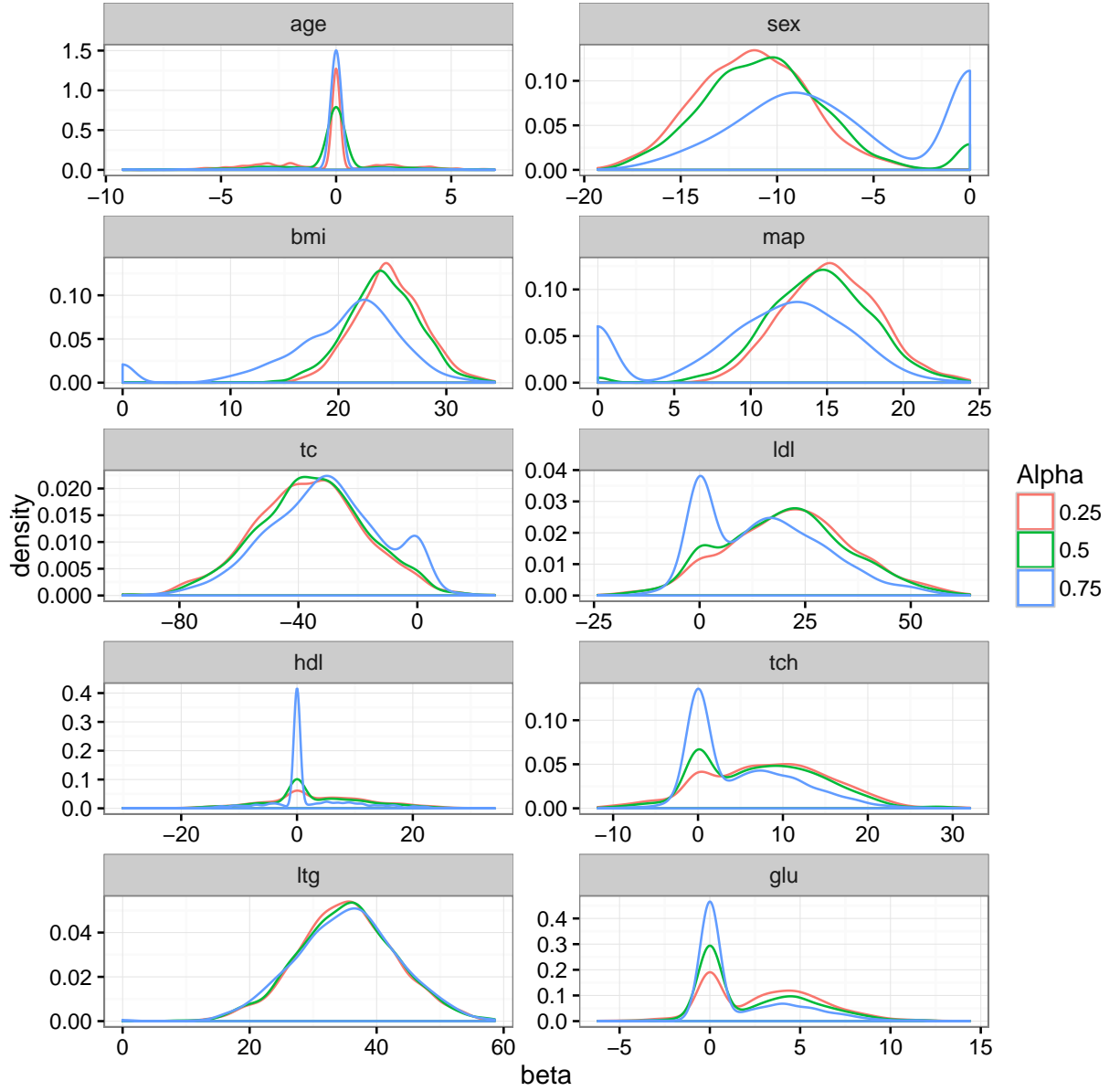
Figure 3: Diabetes example: the weighted Bayesian Bootstrap using different α's are used to draw from the marginal posteriors for β_j's, j = 1,2,...10.
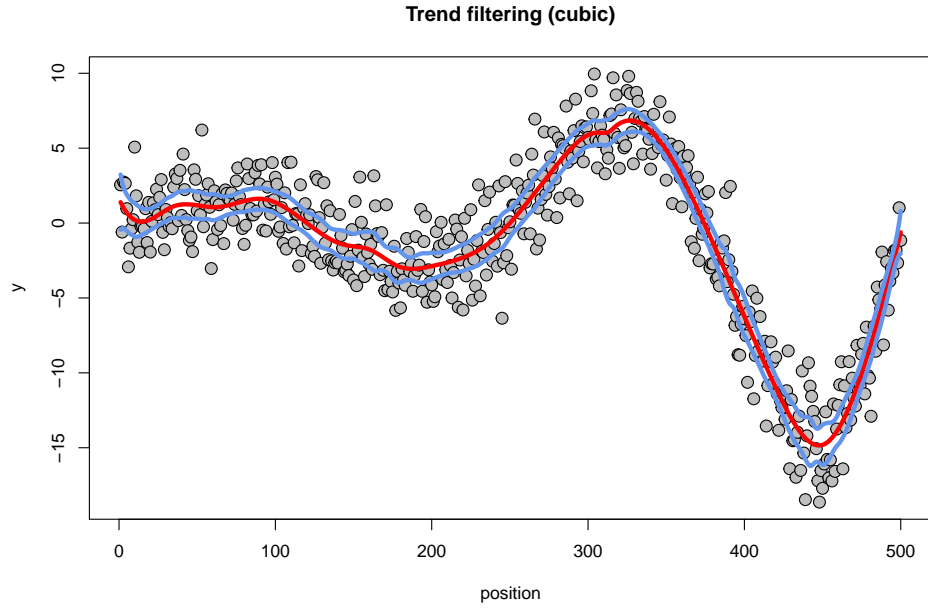
Trend filtering (cubic)

Figure 4: 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 easily computed by WBB. $\lambda = 1000$.
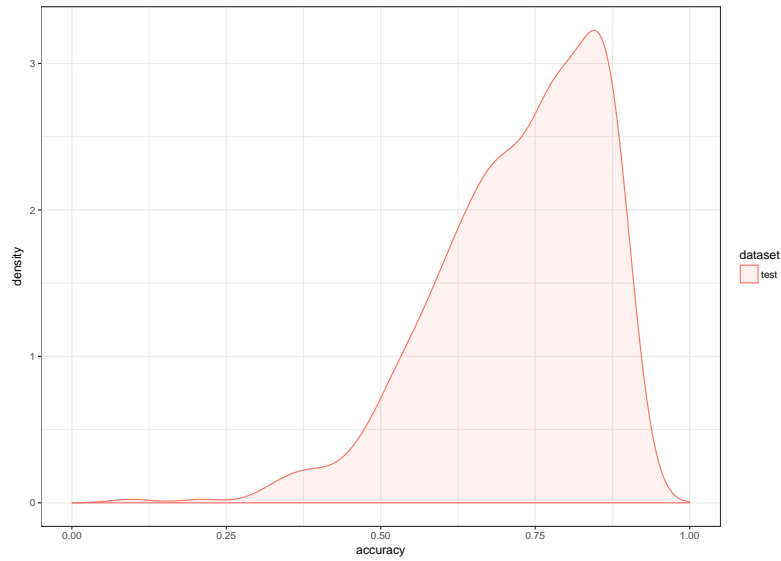

Figure 5: Posterior distribution of the classification accuracy. $n = 500, \lambda = 10^{-4}$.

17