PHILOSOPHICAL TRANSACTIONS A

rsta.royalsocietypublishing.org

Research

Article submitted to journal

Keywords:
Bootstrap, Likelihood-free Inference, Generative Networks

Author for correspondence:
Veronika Ročková
e-mail:
Veronika.Rockova@chicagobooth.edu

Deep Bootstrap for Bayesian Inference

Lizhen Nie[1] and Veronika Ročková[2]

[1]Lizhen@uchicago.edu
[2]Veronika.Rockova@chicagobooth.edu

For a Bayesian, the task to define the likelihood can be as perplexing as the task to define the prior. We focus on situations when the parameter of interest has been emancipated from the likelihood and is linked to data directly through a loss function. We survey existing work on both Bayesian parametric inference with Gibbs posteriors as well as Bayesian non-parametric inference. We then highlight recent bootstrap computational approaches to approximating loss-driven posteriors. In particular, we focus on implicit bootstrap distributions defined through an underlying push-forward mapping. We investigate independent, identically distributed (iid) samplers from approximate posteriors that pass random bootstrap weights through a trained generative network. After training the deep-learning mapping, the simulation cost of such iid samplers is negligible. We compare the performance of these deep bootstrap samplers with exact bootstrap as well as MCMC on several examples (including support vector machines or quantile regression). We also provide theoretical insights into bootstrap posteriors by drawing upon connections to model mis-specification.

1. Introduction

While Bayesian's obligation to specify a prior has been challenged the most, the obligation to specify the likelihood is perhaps even more consequential. Implicit in the Bayesian paradigm is the assumption that a probabilistic model can be formulated which links parameters with data. When constructing such a model, however, one has to reconcile bias implications of model mis-specification (Grünwald and Van Ommen (2017); Müller (2013)). Very often, the primary aim is not modeling the data but rather estimating a parameter. Examples include M-estimators (Huber and Ronchetti (2009)) or other extremum estimators such as censored quantile regression (Powell (1986)), instrumental and robust median regression (Mood (1950)), nonlinear IV and GMM (Hansen and Singleton (1982)). For statistical inference, one may wish to obtain a post-data density summary of the parameter of interest rather than just a point estimate. Bayesian prior-to-posterior inference can be carried out even when one is reluctant about committing to a particular generative model.

THE ROYAL SOCIETY PUBLISHING

One way to liberate the inferential parameter θ_0 from the likelihood is by relating it to data through a general loss function. In medicine, for instance, minimal clinically important difference (Syring and Martin (2017)) or boundary detection in image analysis (Syring and Martin (2020b)) can be formulated as loss minimization problems. It is then possible to perform coherent Bayesian-style updating of prior beliefs, expressed in the prior $\pi(\theta)$, through the so-called Gibbs posteriors (Bissiri et al. (2016); Zhang (2006a,b)). This is a parametric generalization of classical Bayesian inference where the loss function is converted into a pseudo-likelihood function. Another way of expressing uncertainty about θ_0 is through a prior $\pi(F)$, as opposed to the prior $\pi(\theta)$, on the unknown data generating distribution function F_0. Such non-parametric Bayesian inference (Chamberlain and Imbens (2003); Lyddon et al. (2019)) is based on the posterior of F rather than θ. We revisit both the non-parametric and parametric Bayesian approach (with Gibbs posteriors) to inference about parameter targets defined through loss functions.

Recalling the optimal information processing interpretation of the Bayes' rule (Knoblauch et al. (2019); Zellner (1988)), this paper surveys various generalizations of Bayesian inference (including variational inference (Jordan et al. (1999); Wainwright et al. (2008)) and Gibbs posteriors (Catoni (2004); Zhang (2006b))) under one unifying hat. In particular, we adopt the optimization-centric point of view on Bayes' rule which allows a re-interpretation of Bayesian inference as regularized optimization. Any commitment to a Bayesian posterior is a commitment to a particular optimization objective parametrized by the prior, the loss (log-likelihood) function and the class of post-data inferential densities (Knoblauch et al. (2019)). For example, variational Bayes forces the posterior belief into a specific parametric form, transforming the optimization from an infinite-dimensional into a finite-dimensional one. Gibbs posteriors, on the other hand, force priors and data into an exponentially additive relationship through loss functions. Alquier et al. (2016) combined the two by providing a VB computational alternative for Gibbs posteriors.

The repertoire of sampling methods for computing Gibbs posteriors depends on the availability of closed-form conditionals or computational resources. For example, classical MCMC sampling (using e.g. Metropolis-style samplers) may incur large computational costs (Johndrow et al. (2020); Quiroz et al. (2018)). As an alternative, this work investigates the recently proposed generative bootstrap sampler (Shin et al. (2020)) in the context of Bayesian simulation. This generative sampler is trained by learning a deterministic (deep learning) mapping between bootstrap weights and parameters to perform iid sampling. The iid aspect is particularly appealing because, after the mapping has been trained, the simulation cost is negligible compared to sequential samplers. We tailor the strategy of Shin et al. (2020) to the context of Bayesian simulation from approximate (1) Gibbs posteriors for parametric Bayesian inference, and (2) non-parametric Bayesian posteriors. The goal is to learn an implicit distribution prescribed by a deterministic mapping that filters out bootstrap weights to produce samples from an approximate posterior. Implicit distributions have been loosely defined as distributions whose likelihoods are unavailable but which can be sampled from (Li and Turner (2018); Mohamed and Lakshminarayanan (2016)). While implicit distributions have been deployed for Bayesian computation before in the context of variational Bayes (Ruiz and Titsias (2019)), we explore implicit bootstrap distributions to generate samples from approximate posteriors.

Our main purpose is to (1) highlight several recent developments in loss-based Bayesian inference, and to (2) draw attention to generative samplers which are potentially very promising for Bayesian simulation. We investigate their benefits as well as limitations. We found that generative samplers are particularly advantageous when sequential sampling (from conditionals) for each MCMC step and/or when individual optimization for each bootstrap step is costly. We start off by describing loss-based Bayesian inference in Section 2. Section 3 is dedicated to an overview of bootstrap techniques for Bayesian computation. Section 4 provides some theoretical insights and Section 5 describes the performance of the generative bootstrap sampler for Bayesian inference in some classical examples.

The following notation will be used throughout the manuscript. We denote $[n]$ as the set $\{1, 2, \cdots, n\}$, $\mathbf{1}_p \in \mathbb{R}^p$ as a vector of all 1's. We denote with $\|\cdot\|$ the Euclidean norm. For X, a random variable on a probability space (Ω, Σ, P), and f, a function on Ω, we denote $\mathbb{P}(A) = \int_A dP$ for any $A \in \Sigma$, $\mathbb{E}[f(X)] = \int_\Omega f dP$, and also $Pf = \int_\Omega f dP$ to emphasize the underlying probability measure P.

2. Setting the Stage

Assume that we have observed a sequence of iid observations $X^{(n)} = (x_1, \ldots, x_n)'$ from an unknown sampling distribution F_0, i.e. $x_i \sim F_0$. Bayesian inference traditionally requires the knowledge of the true underlying model for F_0. This essentially boils down to specifying a family of likelihood functions $\mathcal{F}_\Theta = \{p_\theta^{(n)}(X^{(n)}) : \theta \in \Theta\}$ indexed by an inferential parameter $\theta \in \Theta \subseteq \mathbb{R}^p$. When there is uncertainty about the parametric family $\mathcal{F}_\Theta$ and mis-specification occurs, likelihood-based inference can be misleading (Grünwald and Van Ommen (2017))). Without the obligation to construct a probabilistic model, it may often be easier to infer about a target parameter θ that is directly tied to F_0, such as the mean, median or other quantile. In other words, one can merely express an interest in some functional of F_0 as opposed to parameter attached to a particular model $p_\theta^{(n)}(X^{(n)}) = \prod_{i=1}^n p_\theta(x_i)$. Allowing $\mathcal{F}_\Theta$ to be unknown, Bissiri et al. (2016) formalized a Bayesian framework for rational updating of beliefs by connecting data to parameters of interest via loss functions. We revisit their development in the context of optimization-centric Bayesian inference.

(a) Optimization-Centric Bayes

Bayesian learning from experience and evidence typically processes information from two sources: (1) a prior density $\pi(\theta)$ which extracts domain expertise, and (2) a likelihood function $p_\theta^{(n)}(X^{(n)})$ which distills the data. Famously, the Bayes' rule produces a post-data density summary for the parameters in a form of a posterior distribution $\pi(\theta \mid X^{(n)})$. There is, however, a conceptually different path for arriving at the posterior. Dating back to at least Csiszár (1975) and Donsker and Varadhan (1975), Bayes' theorem can be interpreted as the optimal information processing rule solving an infinite-dimensional optimization problem (Zellner (1988)). Through the optimization-centric lens (Knoblauch et al. (2019)), Bayesian inference is viewed as regularized optimization

$$\mathcal{L}(\ell; D; \Pi(\Theta)) \equiv \arg \min_{q \in \Pi(\Theta)} \left\{ \mathbb{E}_q \left[\sum_{i=1}^n \ell(\theta; x_i) \right] + D(q \| \pi) \right\} \tag{2.1}$$

indexed by a triplet of parameters: (1) a loss function ℓ, (2) a discrepancy function $D(\cdot\|\cdot)$ gauging the departure from the prior, and (3) a class of probability distributions $\Pi(\Theta)$ to optimize over. The classical likelihood-based Bayesian inference yields

$$\pi(\theta \mid X^{(n)}) = \mathcal{L}\left(-\log \pi_\theta(x); KL; \Pi(\Theta)\right) \tag{2.2}$$

where $\Pi(\Theta)$ is unconstrained and where KL stands for the Kullback-Leibler divergence. The solution of the optimization problem in (2.2) can be rewritten more transparently as (see proof of Theorem 1 in Knoblauch et al. (2019) or Csiszár (1975))

$$\pi(\theta \mid X^{(n)}) = \arg \min_{q \in \Pi(\Theta)} KL \left[q \;\middle\|\; \pi(\theta) \exp\left\{ -\sum_{i=1}^n \ell(\theta; x_i) \right\} Z^{-1} \right], \tag{2.3}$$

where $Z = \int_\theta \exp\{-\sum_{i=1}^n \ell(\theta; x_i)\} \pi(\theta) d\theta$ is the norming constant (assumed to be finite). From (2.3) it can be seen that the optimization problem (2.1) forces priors and losses (log-likelihoods) into an exponentially additive relationship. If the KL term was absent from (2.2), the solution would be a Dirac measure concentrated at the MLE estimator. The incorporation of the prior $\pi(\theta)$ through the KL term in (2.1) allows one to obtain post-data densities for parameters in order to quantify uncertainty and to perform belief updating. The formula (2.3) is a template for generating belief distributions in more general situations, as will be seen below, through information theory.

The information-theoretic representation (2.1) reveals that committing to any particular Bayesian posterior is equivalent to committing to a particular optimization problem determined by (a) the loss function ℓ, (b) the discrepancy metric D, and (c) the space of probability measures $\Pi(\Theta)$. Knoblauch et al. (2019) presents various generalizations of Bayesian inference by altering the parameters $(\ell; D; \Pi(\Theta))$ of the optimization objective in (2.1). For example, constraining the class of distributions $\Pi(\Theta)$ to some parametric form is equivalent to the variational Bayes approach (Wainwright et al. (2008)). Alternatively, replacing the self-information loss $\ell(\theta; x) =$

$-\log \pi_\theta(x)$ with any other loss function, one obtains the so called Gibbs posteriors (more below). Following Knoblauch et al. (2019), we regard (2.1) as the unifying hat behind various generalized Bayesian inference methods.

(b) Bayesian Inference with Gibbs Posteriors

Sometimes, the parameter interest θ is defined indirectly through a general loss function $\ell(\theta; x)$ rather than the likelihood function. For an unknown distribution function F_0, from which we observe an iid vector $X^{(n)}$, one can define such inferential target θ_0 as the minimizer of the average loss (Bissiri et al., 2016)

$$\theta_0 = \theta(F_0) \equiv \arg\min_{\theta\in\Theta} \int \ell(\theta; x) dF_0(x). \tag{2.4}$$

Replacing F_0 with the empirical distribution function P_n of $X^{(n)}$, one obtains an empirical risk minimizer, for example an M-estimator (Huber, 1981; Huber and Ronchetti, 2009; Maronna et al., 2006). In econometrics, extremum estimators (e.g. censored quantile regression of Powell (1986) or instrumental and robust median regression Mood (1950)) are also defined as maximizers of a random finite-sample criterion function whose population counterpart is maximized uniquely at some point $\theta_0 \in \Theta$. In these examples, θ_0 generally cannot be understood as a model parameter but rather as a solution to an optimization (loss minimization) problem. Chernozhukov and Hong (2003) note that implementing such estimators can be challenging and introduce quasi-Bayesian estimators for estimation and inference. Bissiri et al. (2016) propose a related general framework for updating belief distributions as a Bayesian extension of M-estimation. Indeed, even when the parameter cannot be directly assigned any particular model interpretation, it is still possible to perform Bayesian belief updating and uncertainty quantification.

In particular, Bissiri et al. (2016) suggest a decision-theoretic representation of beliefs about θ via a composite loss function over probability measures which gauges fidelity to data and departure from the prior (Berger (1993)). Curiously, their loss function corresponds to the optimization objective in (2.1) where the log-likelihood has been replaced by a general loss function $\ell(\theta; x)$ and where $D(\cdot\|\cdot)$ is the KL divergence. Similarly as in Section (a), it can be shown that the optimal distribution which minimizes this cumulative loss function (without constraining $\Pi(\Theta)$) has an exponentially additive form

$$\tilde{\pi}(\theta \mid X^{(n)}) = \frac{\pi(\theta) \exp\left\{-\alpha \sum_{i=1}^n \ell(\theta; x_i)\right\}}{\int_\Theta \pi(\theta) \exp\left\{-\alpha \sum_{i=1}^n \ell(\theta; x_i)\right\} d\theta}, \tag{2.5}$$

where $\alpha = 1$. Other values of $\alpha > 0$ have been considered to regulate the speed of the learning rate (see Grünwald (2012); Holmes and Walker (2017)). The distribution (2.5) is the "quasi-Bayesian" posterior introduced in Chernozhukov and Hong (2003) and it became known as the Gibbs posterior, a probability distribution for random estimators defined by an empirical measure of risk (Catoni (2004); Zhang (2006b)). The inferential object (2.5) now does not have the interpretation of the usual posterior but rather an optimal prior-to-posterior updating distribution that satisfies coherence and information preservation requirements.

The motivation for Gibbs posteriors can be traced back to the early work of Laplace (Laplace (1774)) who regarded a transformation of a least square criterion function as a statistical belief and obtained point estimates of that distribution "without any assumption about the error distribution" (Stigler (1975)). In thermodynamics, the risk is interpreted as an energy function. In the PAC-Bayesian approach (McAllester (1998); Shawe-Taylor and Williamson (1997)), the Gibbs distribution appears as the probability distribution that minimizes the upper bound of an oracle inequality on the risk of estimators. Estimators derived from Gibbs posteriors, such as quasi-Bayesian mean or median (Chernozhukov and Hong, 2003; McAllester, 2003; Raghunathan, 1993; Rodrigues et al., 1997; Stone, 1965), usually show excellent performance and yet their actual implementation can be challenging. The usual recommendation (Alquier and Biau (2013); Dalalyan and Tsybakov (2012)) is to sample from a Gibbs posterior using MCMC (see e.g. Green et al. (2015) or Ridgway et al. (2014) who propose tempering sequential Monte Carlo which may be too slow for practical use). Alquier et al. (2016) propose a variational Bayes approximation which can achieve the same rate of convergence as the actual Gibbs posterior and which has a polynomial time complexity in convex problems. Our work explores Bootstrap techniques and generative bootstrap samplers for approximating Gibbs posteriors,

going beyond the development in Lyddon et al. (2019). Before delving into the implementation aspects, we distinguish the parametric inference approach with Gibbs posteriors (Bissiri et al. (2016); Chernozhukov and Hong (2003)) from the non-parametric Bayesian learning approach (Chamberlain and Imbens (2003); Lyddon et al. (2019)).

(c) Bayesian Non-parametric Learning

Gibbs posteriors, usually with θ taken from a parametric family, lock priors and losses in an exponentially additive relationship in order to achieve coherent Bayesian updating of beliefs. The uncertainty about the inferential target is a-priori represented in the prior distribution $\pi(\theta)$. The Bayesian non-parametric learning (NPL) approach (Chamberlain and Imbens (1996); Lyddon et al. (2019)), on the other hand, expresses uncertainty about θ through a prior on the unknown distribution function F_0.

Defining the parameter of interest as a functional of F_0 (as in (2.4)), the focus is shifted from θ_0 to F_0. Lyddon et al. (2019) propose a two-step Bayesian learning process by assigning a Dirichlet process (DP) prior $F \sim DP(\alpha, F_\pi)$ on the unknown distribution function F. The base measure F_π conveys prior knowledge about the sampling distribution F_0 and, indirectly, also the parameter θ_0. In the first step, Bayesian non-parametric learning is used to form beliefs about the joint nonparametric density of data and then draws of the non-parametric density are made to repeatedly compute the extremum parameter of interest. Lyddon et al. (2019) call this sampling procedure the loss-likelihood bootstrap. Chamberlain and Imbens (1996) first introduced this strategy using a particular DP posterior (supported only on observations $X^{(n)}$ with Dirichlet-distributed probabilities for each state) that corresponds to the Bayesian bootstrap (Rubin (1981)).

A suitable choice for the base measure F_π is the empirical distribution of the historical data. The second component of the DP prior the concentration parameter $\alpha > 0$ which can be re-interpreted as the effective sample size from the prior F_π. Assigning a DP prior on F_0, the posterior $F \mid X^{(n)} \sim DP(\alpha + n, G_n)$ is also DP with the base measure updated as $G_n = \frac{\alpha}{\alpha+n} F_\pi + \frac{1}{\alpha+n} \sum_{j=1}^{n} \delta_{x_i}$. The posterior for the inferential target θ is then determined from this posterior through the mapping (2.4). In particular, under the stick-breaking representation (Sethuraman (1994)) of the DP posterior, draws from $F \mid X^{(n)}$ are almost surely discrete and the parameter of interest, for each F, can be computed as

$$\theta(F) = \arg\min_{\theta} \sum_{k=1}^{\infty} w_k \times \ell(\theta; y_k)$$

where w_k's are the stick-breaking beta-products and where y_k are iid samples from G_n. Drawing F from the DP posterior requires infinite time when F_π is continuous. Fong et al. (2019) suggest an approximate sampling scheme based on a truncation approximation of F_π and bootstrap-style sampling. The idea is to generate T fake data points $\tilde{x}_j$ from F_π and assign each one a random weight $\tilde{w}_j$. The weights $\boldsymbol{w} = (w_1, \ldots, w_n)'$ for the observed data and fake data $\tilde{\boldsymbol{w}} = (\tilde{w}_1, \ldots, \tilde{w}_T)'$ are jointly sampled from a Dirichlet distribution $Dir(1, \ldots, 1, \alpha/T, \ldots, \alpha/T)'$. The posterior sample of the parameter $\theta(F)$ is then obtained by minimizing an augmented objective $\sum_{i=1}^{n} w_i \ell(x_i; \theta) + \sum_{j=1}^{T} \tilde{w}_j \ell(\tilde{x}_j; \theta)$. Bayesian bootstrap is obtained as a special case when $\alpha = 0$.

The non-parametric Bayesian learning approach (with Bayesian bootstrap (Rubin (1981)) or the loss-likelihood bootstrap (Lyddon et al. (2019))) requires numerous re-computations of the extremum estimates in order to construct the posterior distribution over the parameter of interest. This can be prohibitively slow for optimization problems that are costly to solve. In this work, we investigate the possibility of deploying the generative Bootstrap sampler of Shin et al. (2020) that learns a deterministic mapping of weights $(\boldsymbol{w}, \tilde{\boldsymbol{w}})$ to obtain samples of $\theta(F)$.

3. Deep Bootstrap for Bayesian Computation

Sampling methods are innate to Bayesian computation. Parallelizable iid sampling (with Approximate Bayesian Computation (Beaumont et al. (2002); Pritchard et al. (1999)) or bootstrap (Newton et al. (2021); Newton and Raftery (1994)) has certain advantages over sequential sampling with MCMC. ABC techniques are useful when the likelihood is easier to sample from than to evaluate. Bootstrap-style samplers (Efron (2012); Newton et al. (2021); Newton and Raftery

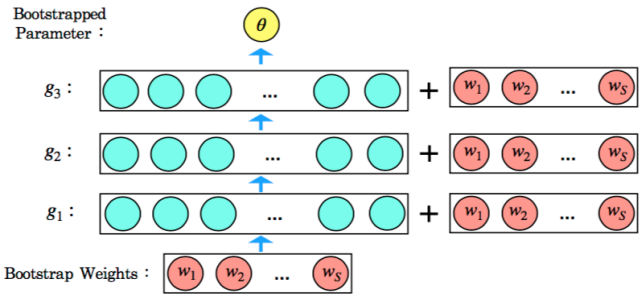

Figure 1: Prototype deep learning architecture from Shin et al. (2020).

(1994); Nie and Ročková (2022); Rubin (1981); Shin et al. (2020)), on the other hand, are beneficial when optimization is easier than, for example, sampling from conditionals. Below, we review recent developments in bootstrap-style posterior computation for Bayesian inference with loss functions.

Weighted likelihood bootstrap (WLB) of Newton and Raftery (1994) is a method for approximately sampling from a posterior distribution of a well-specified parametric statistical model. Samples are generated by computing randomly-weighted maximum likelihood estimates with the weights drawn from a suitable Dirichlet distribution. Such bootstrap samples are obtained as minimizers of randomly re-weighted objective functions, e.g.

$$\widehat{\theta}_{\boldsymbol{w}} = \arg\min_{\theta} \left\{ \sum_{i=1}^{n} w_i \times \ell(\theta; x_i) - \lambda \log \pi(\theta) \right\} \tag{3.1}$$

for a given set of weights $\boldsymbol{w} \in \mathbb{R}_+^n$ drawn from some distribution H and for the log-likelihood loss function $\ell(\cdot; x_i)$. The WLB method sets $\lambda = 0$ in Equation (3.1) and it is not an exact method. However, Newton and Raftery (1994) show that the WLB (weighted likelihood bootstrap with $\lambda = 0$ in (3.1)) is first-order correct (i.e. having the same limiting distribution as the Gibbs posterior with the log-likelihood loss). Edgeworth expansions in Section 4 of Newton and Raftery (1994) reveal that WLB yields a close approximation to the posterior when one uses square of the Jeffrey's prior. For other priors, the authors point out that one may want to "modify the weight distribution to incorporate model and prior information" and that "no general recipe yet exists". As noted in Lyddon et al. (2019), the prior can also be introduced through pseudo-samples. Alternatively, Newton et al. (2021) added a log-prior penalty to incorporate the prior with $\lambda > 0$. Conceivably, one can consider any general loss function $l(\cdot; y_i)$, not just log-likelihood, and use alike strategy to sample from approximate Gibbs posteriors. We illustrate this in a Bayesian least absolute deviation regression example in Section (b).

The WLB sampler can be in principle used to construct approximate quasi-posteriors (Chernozhukov and Hong (2003)), where the prior may need to be incorporated in some form either through distribution H or through a penalty term in (3.1)). This aims at parametric-type inference with Gibbs posteriors (as described in Section (b)). Loss-likelihood bootstrap (Lyddon et al. (2019)) reinterprets WLB as a sampler from an exact posterior over a parameter of interest defined through a loss function under an unknown sampling distribution. This aims at non-parametric Bayesian learning (as described in Section (c)). This is also the strategy pursued in Chamberlain and Imbens (2003) based on the Bayesian bootstrap (Rubin (1981)).

Recall that the loss-likelihood bootstrap of Lyddon et al. (2019) generates samples

$$\widehat{\theta}_{\boldsymbol{w}} = \theta(F_j) \quad \text{where} \quad F_j = \sum_{i=1}^{n} \delta_{X_i} w_i \tag{3.2}$$

with w_i's arriving from, e.g., a Dirichlet distribution and where $\widehat{\theta}_{\boldsymbol{w}}$ can be viewed as a minimizer of an expected loss under F_j as defined in (2.4). At a more intuitive level, each $\widehat{\theta}_{\boldsymbol{w}}$ in (3.1) can be seen as a flexible functional of $\boldsymbol{w}$. This suggests a compelling possibility of treating the distribution of $\widehat{\theta}_{\boldsymbol{w}}$ (be it an approximation to the Gibbs posterior or the non-parametric Bayesian posterior) as an implicit distribution. A distribution is implicit when it is not possible to evaluate its density

Algorithm 1 : Deep Bootstrap Sampler for Gibbs Posterior Learning

Data: Data $\{x_i, 1 \leq i \leq n\}$, number of training epochs T, number of points to sample N, number of Monte Carlo samples K, prior π, learning rate η, number of subgroups S for observed data, $n_S = n/S$ the size of each subgroup.

Result: $\theta^i, i = 1, 2, \cdots, N$.

Training stage:

Initialize weights ϕ **of the fitted function** $\hat{G}$.
for $t = 1, 2, \cdots, T$ **do**
  for $k = 1, 2, \cdots, K$ **do**
    Draw $\tilde{w}^k_{1:S} \sim S \times Dir(1, \cdots, 1)$.
    Set $w^k_{(j-1)n_S+m} = \tilde{w}^k_j$ for all $m = 1, 2, \cdots, n_S$, $j = 1, 2, \cdots, S$.
  end
  Update $\phi \leftarrow \phi - \eta \partial_\phi \left[\sum_{k=1}^K \left[\sum_{j=1}^n w^k_j l\left(G(\tilde{w}^k_{1:S}); x_j\right) + \log \pi\left(G(\tilde{w}^k_{1:S})\right)\right]\right]$.
end

Sampling stage:

for $i = 1, 2, \cdots, N$ **do**
  (2.1) Draw $\tilde{w}^k_{1:S} \sim S \times Dir(1, \cdots, 1)$.
  (2.2) Evaluate $\theta^i = \hat{G}_\phi(\tilde{w}^k_{1:S})$.
end

but it is possible to draw samples from it. One typical way to draw from an implicit distribution is to first sample a noise vector and then push it through a deep neural network (Mohamed and Lakshminarayanan (2016)). Implicit distributions have been deployed within variational Bayes (Pequignot et al. (2020); Ruiz and Titsias (2019)) to obtain flexible distributional approximations to the posterior. Treating bootstrap distributions implicitly, the generative bootstrap sampler (Shin et al. (2020)) draws samples by learning a flexible mapping which transports weights $\boldsymbol{w}$ onto parameters $\theta(F)$. A similar strategy can be used for the WLB approach of Newton and Raftery (1994) where $\widehat{\theta}_{\boldsymbol{w}}$ is linked to $\boldsymbol{w}$ through (3.1).

Instead of re-computing the optimization problem (3.1) for a freshly drawn set of weights $\boldsymbol{w}$ at each step, Shin et al. (2020) suggest training a generator mapping, say $\widehat{\theta}(\boldsymbol{w})$, which has to be learned only once. This mapping is designed to pass random weights w_j's to yield samples from the bootstrap distribution, a sampling process which has negligible cost once the mapping has been learned. The following Lemma (Theorem 2.1 in Shin et al. (2020)) justifies this line of reasoning.

Lemma 1. *(Shin et al. (2020)) Assume that a function $G : \mathbb{R}^n \to \mathbb{R}^p$ is defined as*

$$\hat{G}(\cdot) = \arg\min_{G \in \mathcal{G}} \mathbb{E}_{\boldsymbol{w}} \sum_{i=1}^n w_i \times \ell(G(\boldsymbol{w}); y_i) \tag{3.3}$$

where $\mathbb{E}_{\boldsymbol{w}}$ is the expectation with respect to $\boldsymbol{w} \sim H$. Moreover, assume that the solution $\widehat{\theta}(\boldsymbol{w})$ defined in (3.1) is unique for each given $\boldsymbol{w} \in \mathbb{R}^n$. Then if $\mathcal{G}$ is rich enough to express any function G we have

$$\widehat{G}(\boldsymbol{w}) = \widehat{\theta}_{\boldsymbol{w}}. \tag{3.4}$$

Proof. The proof is given in Section 2.2 of Shin et al. (2020) and rests on the simple observation that, since $\sum w_i l(\theta; y_i) \geq \sum w_i l(\widehat{\theta}_{\boldsymbol{w}}; y_i)$, one has

$$\mathbb{E}_{\boldsymbol{w}} \sum_{i=1}^n w_i l(\widehat{G}(\boldsymbol{w}); y_i) \geq \mathbb{E}_{\boldsymbol{w}} \sum_{i=1}^n w_i l(\widehat{\theta}_{\boldsymbol{w}}; y_i)$$

which implies that $\widehat{G}(\boldsymbol{w}) = \widehat{\theta}_{\boldsymbol{w}}$.

As a concrete example, let us consider $l(\theta; \boldsymbol{x}_i, y_i) = (y_i - \boldsymbol{x}_i^\top \theta)^2$ where $y_i \sim N(\boldsymbol{x}_i^\top \theta, \sigma^2)$. With universal function approximators $\mathcal{G}$ in Equation (3.3), the solution $\widehat{G}$ will be the weighted least squares solution mapping $\widehat{G} : \boldsymbol{w} \mapsto (X^\top W X)^{-1} X^\top W Y$ where $W = \text{diag}(\boldsymbol{w})$.

The result in Lemma 1 implies an important "isomorphism" between bootstrap weights $\boldsymbol{w}$ and parameters θ which can be exploited for faster computation of belief distributions (Gibbs posteriors in Section (b)) or non-parametric Bayesian learning posteriors (Section (c)). For training $G(\cdot)$, one may want to search within mappings $\mathcal{G}$ that are compositions of non-linear transformations, i.e. deep learning mappings. Due to the expressibility of neural networks (see e.g. Barron (1993))), the neural network estimator $\hat{G}(\cdot)$ can be made arbitrarily close to the optimal mapping that satisfies (3.4). This work uses forward deep learning mappings $\mathcal{G}$, thereby the name Deep Bootstrap Sampler. The input to the Deep Bootstrap Sampler $G(\boldsymbol{w})$ is the weight vector $\boldsymbol{w}$ and (according to Lemma 1) in order to train $\widehat{G}(\boldsymbol{w})$, one does not need to know the true solution $\widehat{\theta}_{\boldsymbol{w}}$. One only needs to know how to evaluate the loss (and its gradient) for each data point y_i and each input weight vector $\boldsymbol{w}$, and then use $\mathbb{E}_{\boldsymbol{w}} \sum_{i=1}^n w_i l(\widehat{G}(\boldsymbol{w}); y_i)$ as the objective function. The expectation will typically be approximated with Monte Carlo integration. Thus, the training of the Deep Bootstrap Sampler can proceed in the same way as a typical stochastic gradient descent optimization.

The nested structure of neural networks allows for gradients to be efficiently evaluated using back-propagation (Hecht-Nielsen (1992); Rumelhart et al. (1986)). Once the gradient is computed, stochastic gradient descent algorithm can be used to update the parameters iteratively. Shin et al. (2020) also suggest a specific neural network architecture which re-introduces the weights at each layer. Such a deep approximating class with L network layers, each with n_l neurons, can be written as

$$
\begin{aligned}
g(\boldsymbol{w}) &= g_L(\boldsymbol{Z}_L) \\
\boldsymbol{Z}_{l+1} &= \{g_l(\boldsymbol{Z}_l), \boldsymbol{w}\} \quad \text{for} \quad 1 \le l \le L-1.
\end{aligned}
$$

where $\boldsymbol{Z}_1 = \boldsymbol{w}$ and where each function $g_l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l+1}}$ squashes a multivariate linear combination of the input variables Z_l by a squashing function T, e.g. the Rectified Linear Unit (ReLU) or a sigmoid function. The parameters of the linear combinations for each layer (i.e. intercept terms and slopes) are encapsulated in a vector parameter $\boldsymbol{\phi}$.

In the neural network architecture proposed by Shin et al. (2020), the number of trainable parameters in the l-th layer is $O((n_l + n)n_l)$. When $n \gg n_l$, this number grows linearly in n and thus, reducing the dimension n of input will significantly improve the computational efficiency. Shin et al. (2020) proposed a subgroup bootstrap strategy, which groups the n weights into S equi-sized blocks and $S \ll n$. The weights in the same group are assigned the same value. The S weights are drawn from some distribution H_α. Then the number of trainable parameters in the l-th layer will be reduced to $O((n_l + S)n_l)$. The subgroup strategy significantly boosts the computational advantage of Deep Bootstrap samplers. The architecture incorporating this strategy is shown in Figure 1. We summarize the Deep Bootstrap sampler for Bayesian parametric learning with Gibbs posteriors in Algorithm 1 and for Bayesian non-parametric learning in Algorithm 2.

Since the first draft of this work, there emerged an updated version of the manuscript Shin et al. (2020). The main difference in the updated arXiv manuscript is that it includes double bootstraps, a new neural net architecture called "Weight Multiplicative MLP" (WM-MLP), and a bootstrapped nonparametric MLE. The WM-MLP is inspired by a Taylor approximation of the first derivative of the weighted loss function (see eq 16). The bootstrap sampler based on the WM-MLP performs better in optimization, and the trained bootstrap distribution achieves more accurate variance.

4. Theory

The quantification of the speed of concentration around the truth (as the sample size goes to infinity) is now a standard way of assessing the quality of posteriors (Ghosal et al. (2000)). As Shalizi (2009) states, such Bayesian asymptotic results are "frequentist license for Bayesian practice". Below, we review recent literature related to our development and provide theoretical support for certain aspects of the weighted likelihood bootstrap using connections to model misspecification.

Algorithm 2 : Deep Bootstrap Sampler for Bayesian NPL

Data: Data $\{x_i, 1 \le i \le n\}$, number of prior pseudo samples n', number of training epochs T, number of points to sample N, number of Monte Carlo samples K, concentration parameter α, learning rate η, number of subgroups S for observed data, number of subgroups S' for prior pseudo data, $n_S = n/S, n_{S'} = n/S'$.

Result: $\theta^i, i = 1, 2, \cdots, N$.

Training stage:

Initialize weights ϕ **of the fitted function** $\hat{G}$.

Approximate the prior by drawing $x'_{1:n'} \overset{\text{iid}}{\sim} F_\pi$.

Create the enlarged (observed + prior pseudo) sample $\{x_1, \cdots, x_n, x'_1, \cdots, x'_{n'}\}$.

for $t = 1, 2, \cdots, T$ **do**

  for $k = 1, 2, \cdots, K$ **do**

    Draw $\tilde{w}^k_{1:(S+S')} \sim (S + S') \times Dir(1, \cdots, 1, \alpha/n', \cdots, \alpha/n')$.

    Set $w^k_{(j-1)n_S+m} = \tilde{w}^k_j$ for any $m \in [n_S]$, $j \in [S]$ and $w^k_{n+(j'-1)n_{S'}+m'} = \tilde{w}^k_{j'+S}$, for any $m' \in [n_{S'}], j' \in [S']$.

  end

  Update $\phi \leftarrow \phi - \eta \partial_\phi \left[\sum_{k=1}^K \left[\sum_{j=1}^n w^k_j l\left(G(\tilde{w}^k_{1:(S+S')}); x_j\right) + \sum_{j=1}^{n'} w^k_{j+n} l\left(G(\tilde{w}^k_{1:(S+S')}k); x'_j\right)\right]\right]$.

end

Sampling stage:

for $i = 1, 2, \cdots, N$ **do**

  (2.1) Draw $\tilde{w}^k_{1:(S+S')} \sim (S + S') \times Dir(1, \cdots, 1, \alpha/n', \cdots, \alpha/n')$.

  (2.2) Evaluate $\theta^i = \hat{G}_\phi(\tilde{w}^k_{1:(S+S')})$.

end

Bhattacharya et al. (2019) studied concentration of the so-called fractional α-posteriors obtained by raising the likelihood to some fixed value $\alpha \in (0, 1)$. Han and Yang (2019) study bootstrap-style posteriors where each observation is raised to a *different random* weight w_i where $\sum_{i=1}^n w_i = n$. Bhattacharya et al. (2019) proved that the fractional α-posteriors concentrate on the so-called α-divergence neighborhoods around the truth. The α-divergence is shown to be a valid divergence measure when $\alpha \in (0, 1)$, and the rate of contraction is inflated by a multiplicative factor $\frac{1}{1-\alpha}$. This line of proof, unfortunately, does not extend easily to our bootstrap-style posteriors.[1] In a related paper, Grünwald and Mehta (2020) provided concentration guarantees under a more general "central condition" where α is allowed to be smaller than some critical α^* which equals one when the model is correctly specified and which may be larger or smaller than one with a log-likelihood loss and mis-specified models or with Gibbs posteriors under general losses. Another merit of the result in Grünwald and Mehta (2020) is that it does not assume boundedness of likelihoods, nor does it involve any testing conditions which can be strong under model misspecification settings. We refer to Grünwald and Mehta (2020); Syring and Martin (2020a) for more concentration-rate results for the actual Gibbs posteriors. A referee pointed out that the results Grünwald and Mehta (2020) could potentially yield alternative results for WLB under weaker conditions.

Our goal is not to necessarily study bootstrap-style posteriors, obtained by raising each observation to a random power w_i, but to to study the distribution of extremal (modal) estimators $\widehat{\theta}_{\boldsymbol{w}}$ defined in (3.1). We call the distribution of $\widehat{\theta}_{\boldsymbol{w}}$ as WLB posteriors. While the α-posteriors keep the weight fixed and the randomness stems from treating θ as a random variable with a prior, we treat $\widehat{\theta}_{\boldsymbol{w}}$ as an estimator where the randomness comes from $\boldsymbol{w}$. In a related paper, Han and Yang (2019) study contraction of weighted posterior distributions incorporating both the randomness of θ (through the weighted posterior distribution under the prior $\pi(\theta)$) and the randomness of $\boldsymbol{w}$ (from the distribution of weights $\pi(\boldsymbol{w})$). Again, this is different from WLB since the only source of randomness, given the data, for WLB is $\boldsymbol{w}$. The concept of weighted posterior distributions in Han

[1] Unless $w_i = 1$ for all i's, there must be some weight $w_i > 1$. The existence of such weights invalidates the upper bound in Bhattacharya et al. (2019), and we find it difficult to define a similar (and valid) divergence measure that could be properly upper bounded.

and Yang (2019), however, can still be useful as an intermediate tool for the WLB concentration rate proof. We pursued two strategies for showing the concentration of $\widehat{\theta}_{\boldsymbol{w}}$. The first one is explained in Appendix A where we set out to show concentration properties of the bootstrap-style posteriors but use a different proving technique from Han and Yang (2019). We regard each weighted posterior, for a given vector $\boldsymbol{w}$, as a mis-specified likelihood and we deploy techniques of Kleijn and van der Vaart (2006) to show posterior concentration around the KL projection. Interestingly, due to the symmetry of re-weighting with certain distributions for $\boldsymbol{w}$, it turns out that the KL projection is in fact the true value θ_0 (see Lemma A.1 in the Supplement). This new mis-specification lens would allows us to establish asymptotic normality of bootstrap posteriors, implying that the posterior mode $\widehat{\theta}_{\boldsymbol{w}}$ concentrates around the KL projection θ_0. Since these results hold only for log-likelihood losses, we have decided to pursue a second strategy which yields a broader result for general loss functions. The following Theorem 4.1 directly establishes that the weighted likelihood bootstrap samples $\widehat{\theta}_{\boldsymbol{w}}$ concentrate θ_0.

We utilize tools for establishing convergence rates for M-estimators (Van der Vaart and Wellner (1996)). We denote with $p_\theta(x) = \exp\{-l(\theta; x)\}$ an exponentiated loss, not necessarily a likelihood, and we show that $\hat{\theta}_{\boldsymbol{w}}$ concentrates around the inferential target θ_0 defined in Equation (2.4).

We denote $\mathcal{M}_\epsilon(\theta_0) = \left\{\log[p_\theta/p_{\theta_0}] : d(\theta, \theta_0) < \epsilon\right\}$, and P_0 the probability measure of X_i's, i.e., $X_i \overset{\text{iid}}{\sim} P_0$. For any function class $\mathcal{F}$, we write its Rademacher complexity with respect to P_0 for the sample size n as $\mathcal{R}_n(\mathcal{F})$, i.e.,

$$\mathcal{R}_n(\mathcal{F}) = P_0^{(n)}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(X_i)\right|\right],$$

where $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = 1/2$ for any $i \in [n]$. Denote $\phi_n(\epsilon) = \sqrt{n}\mathcal{R}_n(\mathcal{M}_\epsilon(\theta_0))$. The following theorem establishes the convergence rate of the posterior modes $\hat{\theta}_{\boldsymbol{w}}$. We allow the prior π to depend on the sample size n, and we thereby write π_n for emphasis.

Theorem 4.1. *Assume that:*

(1) (Existence of a suitable semi-metric) There exists a semi-metric $d(\cdot,\cdot)$ on Θ such that

$$P_0 \log[p_{\theta_0}/p_\theta] \geq d^2(\theta, \theta_0) \quad \textit{for all} \quad \theta \in \Theta,$$

(2) (Bounded loss) $\sup_{\theta\in\Theta, X} |\log p_\theta(X)| < \infty$,

(3) (Weight regularization) The weights satisfy

$$w_i \textit{ i.i.d. with } \mathbb{E}w_i = 1, \|w_i\|_{2,1} = \int_0^\infty \sqrt{P\left(|w_i| > x\right)}dx < \infty, \quad \textit{or} \tag{4.1}$$

$$\boldsymbol{w}_n = (w_1, \cdots, w_n) \sim n \times Dir(c, \cdots, c), \quad \textit{for some fixed } c > 0. \tag{4.2}$$

(4) (Proper growth of Rademacher complexity) There exist constants $C > 0$, $\gamma \in (0, 2)$ such that for all $\epsilon > 0, c > 1$,

$$\phi_n(\epsilon) \leq C\sqrt{n}\epsilon^2, \quad \phi_n(c\epsilon) \leq c^\gamma \phi_n(\epsilon).$$

Then, for any $M_n \to \infty$ as $n \to \infty$, we have

$$P_0^{(n)}\mathbb{P}_{\boldsymbol{w}}\left(d(\hat{\theta}_{\boldsymbol{w}}, \theta_0) > M_n\epsilon_n \middle| X^{(n)}\right) \to 0,$$

where $\epsilon_n \to 0$ satisfies

$$\epsilon_n^{-2}\phi_n(\epsilon_n) \leq \sqrt{n} \textit{ for all } n \quad \textit{and} \quad \sup_{\epsilon\geq\epsilon_n} \frac{\sup_{\theta\in\Theta:\epsilon<d(\theta,\theta_0)\leq 2\epsilon} \log \frac{\pi_n(\theta)}{\pi_n(\theta_0)}}{n\epsilon^2} \to 0. \tag{4.3}$$

Proof. See Appendix B.

Remark 1 (Discussion on Assumption 2)**.** *We note that Assumption 2 says the loss is uniformly bounded for all θ and all X. This is a rather strong assumption, unless we are willing to assume our data X is drawn from some compact space; in many cases we also need to restrict the parameter space to a compact subset in the interior of the original parameter space.*

Remark 2 (Discussion on the convergence rate). *Theorem 4.1 establishes a general (not necessarily $\sqrt{n}$) rate of convergence for the distribution of posterior modes. The first part of Equation (4.3) is similar to previous conclusions for M-estimators (Van der Vaart and Wellner (1996)). It shows that the convergence rate for $\hat{\theta}_{\boldsymbol{w}}$ is driven by the growth of $\mathcal{R}_n(\mathcal{M}_\epsilon(\theta_0))$ around $\epsilon = 0$, which is determined by the richness of the function class $\{\log p_\theta : \theta \in \Theta\}$ around $\theta = \theta_0$. For example, with monotone densities, $\epsilon_n = n^{-1/2}$ and with convex densities in $\mathbb{R}^d$, $\epsilon_n = n^{2/(d+4)}$ for $d \leq 3$, $\epsilon_n = n^{1/4}/[\log(n)]^{1/2}$ for $d = 4$ and $\epsilon_n = n^{1/d}$ for $d > 4$. We refer interested readers to Pollard (1991); Van der Vaart and Wellner (1996) for more examples. The second part of (4.3) is less common, and it says that the convergence rate will also be affected by prior $\pi_n(\theta)$. In particular, a sufficient prior mass has to be put around $\theta = \theta_0$, otherwise the convergence rate will be slowed.*

Remark 3 (Errors from deep learning approximation). *We note that all previous results refer to the actual posterior, not the Deep Bootstrap approximation. In other words, we do not consider the estimation error in obtaining $\hat{\theta}_{\boldsymbol{w}}$ using the deep learning approximations. Theoretically, there always exists a sufficiently large neural network whose approximation error is sufficiently small (Gühring et al. (2020); Hornik et al. (1989)). Thus, if we allow its size to grow at a proper rate, we might show existence of a sequence of networks whose mapping converges at a rate no slower than ϵ_n. The actual estimation error of the trained neural network, however, would need to incorporate the actual optimization of the network.*

5. Deep Bootstrap in Bayesian Practice

This section presents several stereotypical toy examples of inference about parameters determined by loss-functions. We aim to illustrate the potential of the deep bootstrap sampler for Bayesian inference.

(a) Bayesian Support Vector Machines

We first demonstrate the performance of the Deep Bootstrap sampler for Bayesian non-parametric learning (Section (c)) in binary classification tasks. Given data $\{(y_i, \boldsymbol{x}_i) \in \{-1, 1\} \bigotimes \mathbb{R}^p\}_{i=1}^n$ where $\boldsymbol{x}_i$ denotes the covariates of the i^{th} observation with a binary label $y_i \in \{-1, +1\}$, binary classification aims to predict y when given $\boldsymbol{x}$ using the sign of $f(x)$, where $f : \mathbb{R}^p \to \mathbb{R}$ is a function to be learned. Various loss functions have been designed to learn f, including the Support Vector Machine (SVM) loss (Cortes and Vapnik (1995))

$$L(y, f(\boldsymbol{x})) = \max\{0, 1 - yf(\boldsymbol{x})\},$$

and the logistic loss (Pearl and Reed (1920))

$$L(y, f(\boldsymbol{x})) = \log(1 + e^{-yf(\boldsymbol{x})}).$$

Suggested by Rosasco et al. (2004), we choose the SVM loss with a linear f, i.e., we minimize the empirical loss $\sum_{i=1}^n l(\beta, \boldsymbol{\theta}; y_i, \boldsymbol{x}_i)$ with

$$l(\beta, \boldsymbol{\theta}; y_i, \boldsymbol{x}_i) = \max\{0, 1 - y_i\left(\beta + \boldsymbol{x}_i'\boldsymbol{\theta}\right)\}, \tag{5.1}$$

where $\beta \in \mathbb{R}$ is the bias and $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_p)' \in \mathbb{R}^p$ are the regression coefficients. For the DP prior, following Fong et al. (2019), we use the prior centering measure

$$\begin{aligned} &F_\pi(y, \boldsymbol{x}) = F_\pi(\boldsymbol{x})F_\pi(y), \quad \text{where} \\ &F_\pi(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n \delta_{\boldsymbol{x}_i}, \quad \text{with } \delta_{\boldsymbol{x}} \text{ the Dirac delta measure centered at } \boldsymbol{x}, \\ &F_\pi(y) = \text{Bernoulli}(0.5). \end{aligned} \tag{5.2}$$

As discussed in Fong et al. (2019), this choice of F_π assumes that $y, \boldsymbol{x}$ are independent and thus is equivalent to assuming $\boldsymbol{\theta} = \mathbf{0}_p$ a priori, which induces similar effects as shrinking priors on $\boldsymbol{\theta}$ (for example, $\|\boldsymbol{\theta}\|_1$ or $\|\boldsymbol{\theta}\|_2$). Regarding the choice of the concentration parameter α, larger α represents stronger beliefs in the prior. Here, following Fong et al. (2019), we set $\alpha = 1.0$.

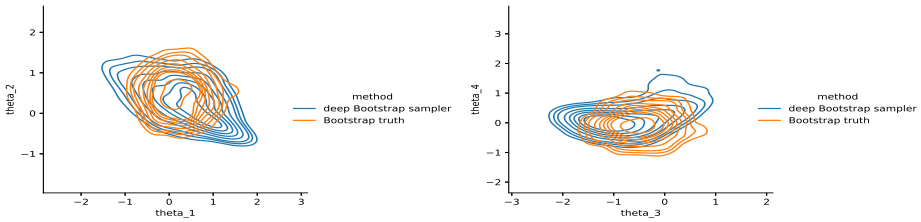

Figure 2: Two-dimensional posterior density plot for θ_i's from deep Bootstrap sampler and the truth for the Bayesian support vector machine example. We set $n = 50, p = 10, \rho = 0.6, \alpha = 1.0$.

To generate simulated data sets, we adopt the setting in Lyddon et al. (2019) and extend to multivariate $\boldsymbol{x}$'s: we sample n i.i.d. data points $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$ from

$$\mathbb{P}(y_i = 1) = \mathbb{P}(y_i = -1) = 1/2, \quad \boldsymbol{x}_i \mid y_i \sim N(y_i \mathbf{1}_p, \Sigma), \tag{5.3}$$

where Σ is a $p \times p$ matrix with all 1's on the diagonal and ρ's off-diagonally. We consider $\rho = 0$ (independent covariates) and a more challenging case $\rho = 0.6$ (equi-correlated covariates). For brevity, results for the independent case is deferred to the Appendix (Section C.3). Note that with Equation (5.3), the inferential target $\theta_0 = (\beta, \boldsymbol{\theta})$ is a solution to the optimization problem defined in Equation (2.4), which does not have a closed form solution for the loss (5.1) and which does not necessarily satisfy $\beta = 0, \boldsymbol{\theta} = \mathbf{1}_p$. For example, when $p = 10, \rho = 0.6$, we have $\beta \approx 0, \boldsymbol{\theta} \approx 0.2 \times \mathbf{1}_p$ by solving (2.4) numerically. The misalignment between the inferential target $\theta_0 = (\beta, \boldsymbol{\theta})$ and the truth $(0, \mathbf{1}_p)$ is not harmful for binary classification tasks as prediction is usually of more interest. We consider varying samples sizes $n \in \{50, 500, 1000, 2000, 5000\}$ and dimensions of covariates $p \in \{10, 50, 100, 200, 500\}$.

We use the deep learning architecture (Figure 1) introduced in Shin et al. (2020) to fit the Deep Bootstrap sampler. As discussed in Shin et al. (2020), the benefit of this structure is that the re-introduction of weights at each hidden layer helps "gradient flow in training deep neural networks" and thus alleviates any potential variance underestimation issues. Using the notation in Section 3, we set $L - 1 = 3$ fully connected (linear function + ReLU) hidden layers, each containing 128 neurons. Our sensitivity analysis in Appendix C.1 shows that the network architecture has an impact on the approximating performance of the Deep Bootstrap sampler, yet such impact is minimal once the network complexity is moderately large (for example, $L - 1 = 3, n_l = 128, l = 1, 2, 3$ sufficies for all experiments we tried). The implementation follows Algorithm 2 and is coded via an optimized machine learning framework `PyTorch` (Paszke et al. (2017)). As suggested by Shin et al. (2020), we use subgroup bootstrap with $S = 100$ and $S' = 10$. The subgroup bootstrap strategy significantly boosts the computational benefit of deep Bootstrap samplers (DBS), and does not hurt much its performance in our experiments. We set the number of Monte Carlo samples $K = 100$, learning rate η initialized at 0.0003 and following a decay rate of $t^{-0.3}$ where t is the current number of epoch (Shin et al. (2020)). In all (n, p)'s we tried, the training usually stabilizes after around 2 000 iterations, and we set $T = 4\,000$ to ensure convergence. RMSprop algorithm (Graves (2013)) is used to update the gradients instead of the vanilla gradient descent in Algorithm 2.

Sampled points $\{\theta^1, \cdots, \theta^N\}$ from the true bootstrap distribution are generated following Algorithm 2 in Fong et al. (2019). It requires solving an optimization problem

$$\theta^j = \arg\max_{(\beta,\boldsymbol{\theta})} \sum_{i=1}^{n} w_i \times l(\beta, \boldsymbol{\theta}; y_i, \boldsymbol{x}_i) + \sum_{i=1}^{n'} w_{i+n} \times l(\beta, \boldsymbol{\theta}; y_i', \boldsymbol{x}_i'), \quad \forall j = 1, 2, \cdots, N, \tag{5.4}$$

with $l(\beta, \boldsymbol{\theta}; y_i, \boldsymbol{x}_i)$ defined in (5.1), $(\boldsymbol{x}_i', y_i') \overset{\text{iid}}{\sim} F_\pi(y, \boldsymbol{x})$ where F_π is defined in (5.2), and the set of weights $(w_1, \cdots, w_{n+n'})$ are drawn from $Dir(1, \cdots, 1, \alpha/n', \cdots, \alpha/n')$. Here n' is the number of pseudo-samples and we set $n' = n$. We solve the optimizatin problem (5.4) using the function `linear_model.SGDClassifier` in Python library `sklearn` (Pedregosa et al. (2011)). This function allows various loss functions and different weight w_i's for each sample point, and optimizes the loss function via stochastic gradient descent. We use the adaptive learning rate schedule (Pedregosa et al. (2011)) and tune the initial learning rate among $\{0.0001, 0.001, 0.01, 0.1\}$. The maximum number of epochs is set to 20 000 with early stopping turned on for saving computations.

Setting		Equi-correlated $\rho = 0.6$						
	metric	accuracy	precision	recall	F1	ROC	PR	time
$p = 10, n = 50$	DBS	0.83	0.82	0.86	0.83	0.91	0.92	48.59+0.58
	WLB	0.86	0.85	0.90	0.87	0.94	0.94	110.99
$p = 50, n = 500$	DBS	0.87	0.88	0.87	0.88	0.94	0.94	97.01+0.65
	WLB	0.86	0.87	0.87	0.87	0.94	0.94	208.10
$p = 100, n = 1000$	DBS	0.86	0.88	0.85	0.86	0.93	0.94	194.80+0.64
	WLB	0.85	0.83	0.85	0.86	0.93	0.94	436.86
$p = 200, n = 2000$	DBS	0.87	0.87	0.87	0.87	0.93	0.93	315.19+0.66
	WLB	0.86	0.86	0.86	0.86	0.93	0.93	1187.21
$p = 500, n = 5000$	DBS	0.89	0.89	0.89	0.89	0.95	0.92	565.05+0.63
	WLB	0.87	0.86	0.87	0.87	0.94	0.93	5677.71

Table 1: Evaluation of approximation properties based on 10 independent runs for Bayesian support vector machine example. DBS standards for 'deep Bootstrap sampler'. WLB standards for samples from the true bootstrap distribution. 'Bias' refers to the l_1 distance of estimated posterior means. 'ROC' refers to the area under the receiver operating characteristic curve; 'PR' refers to the area under the precision-recall curve. The last column in each setting represents the time (in seconds) to generate 10 000 sample points. For deep Bootstrap sampler, time reported is in the form of training time + sampling time.

To evaluate the performance of the Deep Bootstrap sampler, we first consider the two-dimensional density plots of θ_j's. Figure 2 depicts an example where $n = 50, p = 10, \rho = 0.6$. We observe that the Deep Bootstrap sampler captures the bootstrap posterior mean, but less so its variance. This issue has been discussed in Shin et al. (2020). Shin et al. (2020) believe that it is caused by the vanilla feed-forward neural network which prevents variation in the input weights to properly transmit to the output layers as the neural network grows deeper, and they propose the network structure in Figure 1 to alleviate it. Here, we observe that this issue still persists when applied to Bayesian models. The study in Appendix C.2 shows that for Bayesian models, as the sparsity-inducing prior grows stronger, the Deep Bootstrap sampler goes from accurate or slight over-estimation of the variance to more and more severe underestimation. We emphasize that, usually, the variance mismatch issue is not severe as long as the regularization strength is properly selected (e.g., using the BIC criterion (Schwarz (1978)). We discuss the use of BIC in the next example). In this example (with $\alpha = 1.0$), it does not affect the predictive inference on testing data, which is shown in Table 1.

In Table 1, we measure the ability of the Deep Bootstrap sampler to approximate the bootstrap target in terms of predictive performance. For each x_i, we assign an 'average' label by the majority vote based on samples of $(\beta, \boldsymbol{\theta})$. We report quantitative metrics that reflect the quality of this average (accuracy, precision, recall, F1 score). In addition, we report metrics that reflect the shape of the whole Deep Bootstrap posterior (the area under the receiver operating characteristic curve and the precision-recall curve). The metrics are calculated for different choice of n, p's, and on a separate testing set of size 100. The results are summarized in Table 1 which shows that samples generated from the Deep Bootstrap sampler are of high quality for various prediction-related metrics and are close to the target. In terms of predictive performance, the actual bootstrap and DBS are comparable. However, the computing times are dramatically different. We observe that the timing benefit of DBS increases as n or p grow larger.

In summary, this example shows that DBS approximates the true bootstrap posterior well in terms of predictive performance. However, the deep sampler is dramatically faster, especially in large (n, p) settings. The variance mismatch issue mentioned in Shin et al. (2020) exists, but is not fatal. However, as pointed out by one reviewer, one caveat is that the deep learning mapping (similarly as with generative adversarial networks) may not be able to capture multiple modes leading to the so called "mode collapse". In contrast, the independent boostrap samples have a higher chance at capturing multiple modes if the optimization is initialized randomly.

(b) Bayesian LAD Regression

Least squares regression estimators tend to be less robust to outliers or heavy-tailed errors. When robustness becomes a concern, M-estimators (Huber and Ronchetti (2009)) are often used instead of least-square estimators. Given data $\{(y_i, \boldsymbol{x}_i) \in \mathbb{R} \bigotimes \mathbb{R}^p\}_{i=1}^n$ where $\boldsymbol{x}_i$ denotes the covariates of an observation with response y_i, a regression M-estimator is the minimizer of the loss

$$l(\beta, \boldsymbol{\theta}; y_i, \boldsymbol{x}_i) = g(y_i - \beta - \boldsymbol{x}_i^\top \boldsymbol{\theta}). \tag{5.5}$$

where $\beta \in \mathbb{R}$ is the intercept, $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_p) \in \mathbb{R}^p$ is the regression coefficient vector, and $g: \mathbb{R} \to [0, \infty)$ is a residual function that satisfies (i) $g(0) = 0$, (ii) $g(t) = g(-t), \forall t \in \mathbb{R}$, (iii) g is monotonically increasing (Rousseeuw and Leroy (2005)). Statistical literature on robustness has proposed a variety of residual functions g, including the Huber function (Huber (1964))

$$g(t; \delta) = \begin{cases} \frac{1}{2}t^2, & \text{for } |t| \leq \delta \\ \delta\left(|t| - \frac{1}{2}\delta\right), & \text{otherwise} \end{cases}$$

and the absolute value function (Boscovich (1757)) $g(t) = |t|$. Here, we consider the case where g is the absolute value function, i.e., least absolute deviation (LAD) regression (Boscovich (1757)). A penalized LAD regression model is investigated in Lambert-Lacroix and Zwald (2011); Wang et al. (2007); Wang (2013); Wang et al. (2006); Zou (2006), which minimizes

$$\sum_{i=1}^{n} \left| y_i - \beta - \boldsymbol{\theta}^\top \boldsymbol{x}_i \right| + \lambda_0 \sum_{j=1}^{p} |\theta_j|. \tag{5.6}$$

Note that the intercept β is excluded from the penalty term as suggested by Wang et al. (2006). In (5.6), the regularization strength λ_0 may be selected from the classical BIC criteria as recommended by Lambert-Lacroix and Zwald (2011); Schwarz (1978). It was pointed out by a reviewer, however, that despite its use in Lambert-Lacroix and Zwald (2011) the BIC choice of λ_0 may not be the best practice. BIC is theoretically justified only when the model is correctly specified, while in our case the model can be quite mis-specified. As shown in Grünwald and Van Ommen (2017), Bayesian marginal likelihoods can point in a wrong direction. Our goal, however, is not to choose the best λ_0 but rather to compare posterior approximations for a given value λ_0.

Viewing (5.6) from a Bayesian viewpoint, minimizing (5.6) is equivalent to estimating the mode of a Bayesian model with a loss

$$l(\beta, \theta; y_i) = \left| y_i - \beta - \boldsymbol{\theta}^\top \boldsymbol{x}_i \right|, \tag{5.7}$$

and a prior

$$\pi(\boldsymbol{\theta}) = \prod_{j=1}^{p} \left[\frac{\lambda_0}{2} e^{-\lambda_0 |\theta_j|} \right]. \tag{5.8}$$

However, one might be interested in not only the posterior mode but also uncertainty quantification. The Gibbs posterior defined by (2.5) in Section (b) provides one such uncertainty measurement. We compute the Gibbs posterior using MCMC. Another possibility is to obtain approximations, either by directly solving the optimization problem defined in (3.1), or by using the deep Bootstrap sampler (Algorithm 2). This examples investigates these three objects for uncertainty quantification.

We simulate data following the settings in Lambert-Lacroix and Zwald (2011): n i.i.d. data points $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$ are generated from

$$y_i = \beta^* + \boldsymbol{x}_i^\top \boldsymbol{\theta}^* + \sigma \epsilon_i, \quad \boldsymbol{x}_i \overset{\text{iid}}{\sim} N(0, \Sigma), \tag{5.9}$$

where $\beta^* = 1$, $\boldsymbol{\theta}^* = (1.5, 2, 3, 0, 0, \cdots, 0) \in \mathbb{R}^p$, Σ is a Toeplitz matrix with $\Sigma_{i,j} = (0.5)^{|i-j|}$. Following Lambert-Lacroix and Zwald (2011)), we consider two challenging cases with heavy-tailed noise distribution:

- Model 1: large outliers. $\epsilon_i = v_i / \sqrt{\text{var}(v_i)}$, $\sigma = 9.67$, and $v_i \overset{\text{iid}}{\sim} 0.9N(0, 1) + 0.1N(225)$.
- Model 2: sensible outliers. $\epsilon_i = v_i / \sqrt{\text{var}(v_i)}$, $\sigma = 9.67$, and $v_i \overset{\text{iid}}{\sim} Laplace(1)$.

One may show that the solution to (2.4) equals the truth (i.e., $\theta_0 = (\beta, \boldsymbol{\theta}) = (\beta^*, \boldsymbol{\theta}^*)$) under some mild conditions as in Gross and Steiger (1979); Pollard (1991); Ruzinsky and Olsen (1989). We note that this equality holds for both Model 1 and 2. We investigate different choices of $n \in \{8, 10, 20, 50\}$ and $p \in \{100, 200, 500, 1000\}$.

Both the Gibbs posterior (2.5) and bootstrap samples (3.1) use the loss (5.7) and the prior (5.8), where the regularization strength λ_0 in prior (5.8) is set to the one that minimizes the BIC criteria (Schwarz (1978)) among $\log(\lambda_0) \in \{-6, \cdots, 1\}$ (an equi-spaced sequence starting at -6, ending at

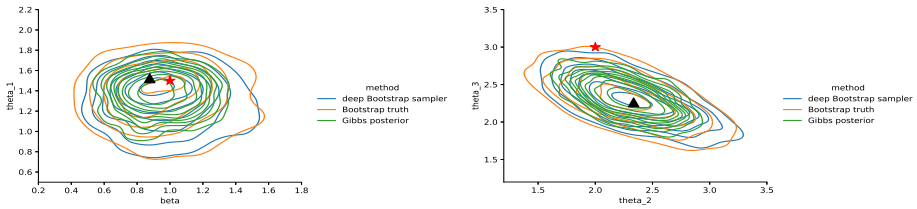

Figure 3: Two-dimensional posterior density plot for θ_i's in Model 1 for Bayesian LAD example. We set $n = 100, p = 8$. The location of true parameters are marked in red star. The location of the minimizer of Equation (5.6) are marked in black triangle.

1 and of length 20), as suggested by Lambert-Lacroix and Zwald (2011). For the Gibbs posterior, we set the learning rate $\alpha = 1$ to match (5.6) which is used to generate bootstrap samples. We note, however, that this choice ($\alpha = 1$) may be suboptimal if one wants to obtain a Gibbs posterior that achieves good statistical properties (Grünwald and Van Ommen, 2017; Grünwald and Mehta, 2020). In addition to this literature, we refer readers to Bissiri et al. (2016) and Lyddon et al. (2019) for more discussion on the problem of determining an optimal α.

In practice, the Gibbs posterior is generated using the Metropolis-Hasting algorithm described in Chernozhukov and Hong (2003). We implement it in Python. We initialize $\theta_0, \boldsymbol{\theta}$ at the mode of the Gibbs posterior solved by `QuantileRegressor`. The proposal density $q(x \mid y)$ is set to the density of $N(y, \sigma' I_p)$. We set $\sigma' = 0.1$ for small data sets and $\sigma' = 0.01$ for large ones as the true Gibbs posterior tends to have smaller variance as the dimension grows. We run MCMC for 1 000 000 iterations and discard the first 10 000 as burn-in. Gelman-Rubin diagnostic (Gelman and Rubin (1992)) computed by R package `coda` (Plummer et al. (2006)) is checked to ensure convergence.

For the Bootstrap samples, we solve the optimization problem (3.1) with $w \sim n \times Dir(1, 1, \cdots, 1)$, loss (5.7), and prior (5.8) using the built-in function `QuantileRegressor`. This function solves (3.1) by linear programming, and we use its default interior-point solver with default parameters. For the Deep Bootstrap sampler, we implement Algorithm 2 using `Pytorch`, with a subgroup size $S = 100$. The other settings, including network architecture, training schedule and hyper-parameters, are identical to the Bayesian support vector machine example.

Let us first consider the posterior densty of θ_j's. Figure 3 shows the two-dimensional posterior density plot in Model 1. Results for Model 2 are very similar and are included in the Appendix C.4. We observe that samples from all three methods are centered around the same location. The center is close to the minimizer of Equation (5.6), which is different from the truth due to shrinkage effects introduced from the prior. The Gibbs posterior tends to have the smallest variance, in contrast to the true Bootstrap samples which has the largest variance.

To quantitatively compare each method, we report the bias, lengths of 90% credible intervals and their coverage for various choice of (n, p)'s in Table 2. We separate between active and inactive coordinates. Table 2 also includes the time cost for each method. For fairness, we compare the time required to generate the same number of effective samples. Since each sample from boostrap is independent, we treat the total number of samples as the effective sample size for both bootstrap samplers. The effective sample size for Gibbs posterior is calculated using R package `coda`. Table 2 shows the deep Bootstrap sampler is much faster than both the true bootstrap and the Gibbs posterior, especially in large datasets.

We conclude that, in this example, DBS reconstructions could be a viable alternative to Gibbs posteriors. The Deep Bootstrap sampler achieves a similar bias and a larger variance, but is much faster than the Metropolis-Hastings algorithm.

6. Discussion

This paper surveys several recent contributions to the Bayesian literature on learning about parameters defined by loss functions. We highlighted a new promising direction for Bayesian computation using generative bootstrap. We demonstrated the potential of this new strategy on several examples. This paper aims to draw practitioners' attention towards posterior sampling techniques beyond the traditional MCMC technology. Since bootstrap is inherently parallelizable, the generative sampler might be less favorable when individual optimizations inside each

Setting		Model 1							Model 2						
	metric	coverage		length of 90% CI		bias		time	coverage		length of 90% CI		bias		time
		+	−	+	−	+	−		+	−	+	−	+	−	
$p = 8, n = 100$	DBS	0.90	0.98	1.10	1.23	0.23	0.268	50.51+0.63	0.90	0.88	3.43	3.82	0.79	0.92	57.20+0.76
	WLB	0.98	1.00	1.37	1.52	0.23	0.26	534.08	0.90	0.95	4.19	4.71	0.79	0.90	546.73
	Gibbs	0.78	0.80	0.71	0.80	0.23	0.29	42.71	0.45	0.43	1.22	1.32	0.82	0.96	82.12
$p = 10, n = 100$	DBS	0.85	0.85	0.99	1.15	0.26	0.33	79.83+1.01	0.90	0.83	3.20	3.93	0.77	1.14	68.39+0.85
	WLB	0.95	0.97	1.27	1.44	0.25	0.34	889.63	0.95	0.93	4.11	4.86	0.74	1.18	780.12
	Gibbs	0.75	0.65	0.67	0.76	0.27	0.34	58.58	0.43	0.33	1.14	1.39	0.79	1.16	103.04
$p = 20, n = 200$	DBS	0.90	0.88	0.72	0.76	0.18	0.19	70.05+0.86	0.78	0.82	2.28	2.46	0.66	0.71	74.97+0.94
	WLB	1.00	0.95	0.93	1.00	0.19	0.20	2891.89	0.93	0.91	2.94	3.29	0.69	0.73	3041.20
	Gibbs	0.35	0.35	0.26	0.27	0.19	0.29	696.23	0.43	0.39	0.85	0.93	0.64	0.71	1550.77
$p = 50, n = 500$	DBS	0.78	0.83	0.41	0.43	0.13	0.12	74.43+0.91	0.85	0.82	1.35	1.42	0.42	0.41	81.64+1.01
	WLB	0.93	0.96	0.60	0.63	0.13	0.12	> 6 hours	0.95	0.95	1.89	2.04	0.42	0.42	> 6 hours
	Gibbs	0.68	0.74	0.32	0.34	0.13	0.12	703.52	0.38	0.43	0.54	0.59	0.40	0.41	1529.81
$p = 50, n = 1000$	DBS	0.93	0.80	0.27	0.30	0.08	0.09	75.08+0.87	0.88	0.86	0.86	0.95	0.23	0.25	81.90+0.93
	WLB	0.98	0.91	0.38	0.41	0.15	0.20	> 2 days	0.98	0.94	1.17	1.24	0.23	0.26	> 2 days
	Gibbs	0.73	0.68	0.21	0.23	0.08	0.09	545.54	0.38	0.47	0.35	0.39	0.24	0.25	1140.76

Table 2: Evaluation of approximation properties in different settings based on 10 independent runs of Bayesian LAD regression. DBS standards for 'Deep Bootstrap Sampler'. WLB standards for samples from the true bootstrap distribution. Coverage stands for the empirical coverage of 90% credible intervals. 'Bias' refers to the l_1 distance between estimated posterior means and the truth. We denote with $+$ an average over active coordinates, and with $-$ an average over inactive coordinates. Times reported are the number of seconds (unless otherwise noted) taken to generate 10 000 effective samples for each procedure. For deep Bootstrap sampler, time reported is in the form of training time + sampling time.

bootstrap replication are relatively easy. In such cases, the computational benefit of generative samplers has to be considered in a case-by-case manner, comparing the relative overhead for training such a sampler versus the cost for obtaining one sample point in the alternative approach. Second, as mentioned in Lyddon et al. (2019), the independent sampling nature of Bayesian Bootstrap (Rubin, 1981), weighted likehood bootstrap (Newton and Raftery, 1994) and NPL (Chamberlain and Imbens, 1996; Lyddon et al., 2019) have made them particularly capable of capturing multi-modality in posteriors. Approximating the weight-to-posterior mapping using a deep neural network, however, might weaken this ability.

Funding. The author gratefully acknowledges the support from the James S. Kemper Faculty Fund at the Booth School of Business and the National Science Foundation (Grant No. NSF DMS-1944740).

References

Alquier, P. and Biau, G. (2013). Sparse single-index model. *Journal of Machine Learning Research*, 14(1):243–280.

Alquier, P., Ridgway, J., and Chopin, N. (2016). On the properties of variational approximations of Gibbs posteriors. *Journal of Machine Learning Research*, 17(236):1–41.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945.

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035.

Berger, J. O. (1993). *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media.

Bhattacharya, A., Pati, D., and Yang, Y. (2019). Bayesian fractional posteriors. *The Annals of Statistics*, 47(1):39–66.

Bissiri, P. G., Holmes, C. C., and Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130.

Boscovich, R. J. (1757). De litteraria expeditione per pontificiam ditionem, et synopsis amplioris operis, ac habentur plura ejus ex exemplaria etiam sensorum impessa. *Bononiensi Scientiarum et Artum Instuto Atque Academia Commentarii*, 4:353–396.

Catoni, O. (2004). *Statistical learning theory and stochastic optimization*. Springer Science & Business Media.

Chamberlain, G. and Imbens, G. W. (1996). Nonparametric applications of Bayesian inference. Technical report, National Bureau of Economic Research.

Chamberlain, G. and Imbens, G. W. (2003). Nonparametric applications of Bayesian inference. *Journal of Business & Economic Statistics*, 21(1):12–18.

Chernozhukov, V. and Hong, H. (2003). An mcmc approach to classical estimation. *Journal of Econometrics*, 115(2):293–346.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Csiszár, I. (1975). I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158.

Dalalyan, A. S. and Tsybakov, A. B. (2012). Mirror averaging with sparsity priors. *Bernoulli*, 18(3):914–944.

Donsker, M. D. and Varadhan, S. S. (1975). Asymptotic evaluation of certain Markov process expectations for large time, I. *Communications on Pure and Applied Mathematics*, 28(1):1–47.

Efron, B. (2012). Bayesian inference and the parametric bootstrap. *The annals of applied statistics*, 6(4):1971.

Fong, E., Lyddon, S., and Holmes, C. (2019). Scalable nonparametric sampling from multimodal posteriors with the posterior bootstrap. In *International Conference on Machine Learning*, pages 1952–1962.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472.

Ghosal, S., Ghosh, J. K., and Van Der Vaart, A. W. (2000). Convergence rates of posterior distributions. *Annals of Statistics*, pages 500–531.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Green, P. J., Łatuszyński, K., Pereyra, M., and Robert, C. P. (2015). Bayesian computation: a summary of the current state, and samples backwards and forwards. *Statistics and Computing*, 25(4):835–862.

Gross, S. and Steiger, W. L. (1979). Least absolute deviation estimates in autoregression with infinite variance. *Journal of Applied Probability*, 16(1):104–116.

Grünwald, P. (2012). The safe Bayesian. In *International Conference on Algorithmic Learning Theory*, pages 169–183. Springer.

Grünwald, P. and Van Ommen, T. (2017). Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, 12(4):1069–1103.

Grünwald, P. D. and Mehta, N. A. (2020). Fast rates for general unbounded loss functions: From ERM to generalized Bayes. *J. Mach. Learn. Res.*, 21:56–1.

Gühring, I., Kutyniok, G., and Petersen, P. (2020). Error bounds for approximations with deep relu neural networks in w s, p norms. *Analysis and Applications*, 18(05):803–859.

Han, W. and Yang, Y. (2019). Statistical inference in mean-field variational Bayes. *arXiv e-prints*, arXiv:1911.01525.

Hansen, L. P. and Singleton, K. J. (1982). Generalized instrumental variables estimation of nonlinear rational expectations models. *Econometrica: Journal of the Econometric Society*, 50(5):1269–1286.

Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier.

Holmes, C. C. and Walker, S. G. (2017). Assigning a value to a power likelihood in a general Bayesian model. *Biometrika*, 104(2):497–503.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Statistics*, 53(1):73–101.

Huber, P. J. (1981). *Robust statistics*. Wiley, New York.

Huber, P. J. and Ronchetti, E. M. (2009). *Robust statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, NJ, 2 edition.

Johndrow, J., Orenstein, P., and Bhattacharya, A. (2020). Scalable approximate mcmc algorithms for the horseshoe prior. *Journal of Machine Learning Research*, 21(73).

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kleijn, B. J. and van der Vaart, A. W. (2006). Misspecification in infinite-dimensional Bayesian statistics. *The Annals of Statistics*, 34(2):837–877.

Knoblauch, J., Jewson, J., and Damoulas, T. (2019). Generalized variational inference: Three arguments for deriving new posteriors. *arXiv e-prints*, arXiv:1904.02063.

Lambert-Lacroix, S. and Zwald, L. (2011). Robust regression through the huber's criterion and adaptive lasso penalty. *Electronic Journal of Statistics*, 5:1015–1053.

Laplace, P.-S. (1774). Mémoire sur les suites récurro-récurrentes et sur leurs usages dans la théorie des hasards. *Mémoires de l'Académie Royale des Sciences Paris*, 6:353–371.

Li, Y. and Turner, R. E. (2018). Gradient estimators for implicit models. In *International Conference on Learning Representations*.

Lyddon, S., Holmes, C., and Walker, S. (2019). General Bayesian updating and the loss-likelihood bootstrap. *Biometrika*, 106(2):465–478.

Maronna, R. A., Martin, R. D., and Yohai, V. J. (2006). *Robust statistics: theory and methods*. Wiley Series in Probability and Statistics. John Wiley & Sons, Chichester.

McAllester, D. A. (1998). Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234. ACM Press.

McAllester, D. A. (2003). PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1):5–21.

Mohamed, S. and Lakshminarayanan, B. (2016). Learning in implicit generative models. *arXiv e-prints*, arXiv:1610.03483.

Mood, A. M. (1950). Introduction to the theory of statistics.

Müller, U. K. (2013). Risk of Bayesian inference in misspecified models, and the sandwich covariance matrix. *Econometrica*, 81(5):1805–1849.

Newton, M. A., Polson, N. G., and Xu, J. (2021). Weighted Bayesian bootstrap for scalable posterior distributions. *Canadian Journal of Statistics*, 49(2):421–437.

Newton, M. A. and Raftery, A. E. (1994). Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(1):3–26.

Nie, L. and Ročková, V. (2022). Bayesian bootstrap spike-and-slab LASSO. *Journal of the American Statistical Association*, (just-accepted):1–35.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Pearl, R. and Reed, L. J. (1920). On the rate of growth of the population of the united states since 1790 and its mathematical representation. *Proceedings of the national academy of sciences*, 6(6):275–288.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pequignot, Y., Alain, M., Dallaire, P., Yeganehparast, A., Germain, P., Desharnais, J., and Laviolette, F. (2020). Implicit variational inference: the parameter and the predictor space. *arXiv e-prints*, arXiv:2010.12995.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). Coda: Convergence diagnosis and output analysis for mcmc. *R News*, 6(1):7–11.

Pollard, D. (1991). Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, 7(2):186–199.

Powell, J. L. (1986). Censored regression quantiles. *Journal of econometrics*, 32(1):143–155.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. (1999). Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798.

Quiroz, M., Kohn, R., Villani, M., and Tran, M.-N. (2018). Speeding up mcmc by efficient data subsampling. *Journal of the American Statistical Association*.

Raghunathan, T. (1993). A quasi-empirical Bayes method for small area estimation. *Journal of the American Statistical Association*, 88(424):1444–1448.

Ridgway, J., Alquier, P., Chopin, N., and Liang, F. (2014). PAC-Bayesian AUC classification and scoring. In *Advances in Neural Information Processing Systems*, pages 658–666.

Rodrigues, J., Bolfarine, H., and Cordeiro, G. M. (1997). Nonlinear quasi-Bayesian theory and inverse linear regression. *Communications in statistics-theory and methods*, 26(10):2347–2361.

Rosasco, L., De Vito, E., Caponnetto, A., Piana, M., and Verri, A. (2004). Are loss functions all the same? *Neural computation*, 16(5):1063–1076.

Rousseeuw, P. J. and Leroy, A. M. (2005). *Robust regression and outlier detection*. John wiley & sons.

Rubin, D. B. (1981). The Bayesian bootstrap. *The annals of statistics*, pages 130–134.

Ruiz, F. and Titsias, M. (2019). A contrastive divergence for combining variational inference and mcmc. In *International Conference on Machine Learning*, pages 5537–5545. PMLR.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Ruzinsky, S. A. and Olsen, E. T. (1989). Strong consistency of the lad (l/sub 1/) estimator of parameters of stationary autoregressive processes with zero mean. *IEEE transactions on acoustics, speech, and signal processing*, 37(4):597–600.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, pages 461–464.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica sinica*, 4:639–650.

Shalizi, C. R. (2009). Dynamics of Bayesian updating with dependent data and misspecified models. *Electronic Journal of Statistics*, 3:1039–1074.

Shawe-Taylor, J. and Williamson, R. C. (1997). A PAC analysis of a Bayesian estimator. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 2–9.

Shin, M., Wang, S., and Liu, J. S. (2020). Generative multiple-purpose sampler for weighted m-estimation. *arXiv e-prints*, arXiv:2006.00767.

Stigler, S. M. (1975). Studies in the history of probability and statistics. xxxiv: Napoleonic statistics: The work of Laplace. *Biometrika*, 62(2):503–517.

Stone, M. (1965). Right haar measure for convergence in probability to quasi posterior distributions. *The Annals of Mathematical Statistics*, 36(2):440–453.

Syring, N. and Martin, R. (2017). Gibbs posterior inference on the minimum clinically important difference. *Journal of Statistical Planning and Inference*, 187:67–77.

Syring, N. and Martin, R. (2020a). Gibbs posterior concentration rates under sub-exponential type losses. *arXiv e-prints*, arXiv:2012.04505.

Syring, N. and Martin, R. (2020b). Robust and rate-optimal gibbs posterior inference on the boundary of a noisy image. *The Annals of Statistics*, 48(3):1498–1513.

Van der Vaart, A. and Wellner, J. (1996). *Weak convergence and empirical processes: with applications to statistics*. Springer Science & Business Media.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.

Wang, H., Li, G., and Jiang, G. (2007). Robust regression shrinkage and consistent variable selection through the lad-lasso. *Journal of Business & Economic Statistics*, 25(3):347–355.

Wang, L. (2013). The l1 penalized lad estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 120:135–151.

Wang, L., Gordon, M. D., and Zhu, J. (2006). Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 690–700. IEEE.

Zellner, A. (1988). Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280.

Zhang, T. (2006a). From ε-entropy to kl-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, 34(5):2180–2210.

Zhang, T. (2006b). Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.