Machine Learning and the Future of Bayesian Computation

Steven Winter, Trevor Campbell[†], Lizhen Lin[†], Sanvesh Srivastava[†], and David B. Dunson

Abstract. Bayesian models are a powerful tool for studying complex data, allowing the analyst to encode rich hierarchical dependencies and leverage prior information. Most importantly, they facilitate a complete characterization of uncertainty through the posterior distribution. Practical posterior computation is commonly performed via MCMC, which can be computationally infeasible for high dimensional models with many observations. In this article we discuss the potential to improve posterior computation using ideas from machine learning. Concrete future directions are explored in vignettes on normalizing flows, Bayesian coresets, distributed Bayesian inference, and variational inference.

Key words and phrases: Coresets, federated learning, machine learning, normalizing flows, posterior computation, variational Bayes.

1. INTRODUCTION

There is immense interest in performing inference and prediction for complicated real-world processes within science, industry, and policy. Bayesian models are appealing because they allow specification of rich generative models encompassing hierarchical structures in the data, natural inclusion of information from experts and/or previous research via priors, and a complete characterization of uncertainty in learning/inference/prediction through posterior and predictive distributions. The primary hurdle in applying Bayesian statistics to complex real-world data is posterior computation. In practice, posterior computation evaluating posterior probabilities/expectations, credible intervals for parameters, posterior inclusion probabilities for features, posterior predictive intervals, etc – is typically based on posterior samples using Markov chain Monte Carlo (MCMC). Standard MCMC approaches often fail

Steven Winter: PhD Student, Department of Statistical Science, Duke University (e-mail: steven.winter@duke.edu). Trevor Campbell: Assistant Professor, Department of Statistics, University of British Columbia (e-mail: trevor@stat.ubc.ca). Lizhen Lin: Robert and Sara Lumpkins Associate Professor, Department of Applied and Computational Mathematics and Statistics, University of Notre Dame (e-mail: lizhen.lin@nd.edu). Sanvesh Srivastava: Associate Professor, Department of Statistics and Actuarial Science, University of Iowa (e-mail: sanvesh-srivastava@uiowa.edu). David B. Dunson: Arts and Sciences Distinguished Professor, Departments of Statistical Science and Mathematcs, Duke University (e-mail: dunson@duke.edu).

to converge when the posterior has complicated geometry, such as multiple distant modes or geometric/manifold constraints. Even sampling from simple posteriors can be challenging when the data has tens or hundreds of millions of observations. This article focuses on the future of Bayesian computation, with emphasis on posterior inference for high dimensional, geometrically complicated targets with potentially millions of datapoints.

The recent explosive success of machine learning is key in shaping our vision for the future of Bayesian computation. To make our vision concrete, we have prepared four vignettes covering disjoint cutting-edge computational techniques, all involving ideas from machine learning. The first vignette describes normalizing flows as a new tool for adaptive MCMC with complicated targets; the second describes Bayesian coresets as a method of data compression prior to sampling; the third describes distributed Bayesian inference for huge datasets; the fourth describes modern variational inference for settings where the previous techniques falter. All sections focus heavily on promising avenues for future research.

2. SAMPLING USING DEEP GENERATIVE MODELS

The Metropolis Hastings (MH) algorithm (often within Gibbs) is by far the most popular tool for sampling posterior distributions [33]. Good mixing is critically dependent on how closely the MH proposal distribution mimics the target distribution. Higher dimensional targets with increasingly complicated geometry require increasingly flexible proposal distributions which become difficult to tune. Consequently, it is routine to settle for simpler proposals which

[†]These authors contributed equally.

provide a good local approximation to the target, such as a multivariate Gaussian. Parameters are then tuned to encourage efficient exploration, e.g. by adaptively learning the posterior covariance [53, 139, 153] or by discretizing dynamics driven by the target [106, 86]. A major limitation of local methods is their practical inability to cross low-probability regions, resulting in poor convergence rates for multimodal distributions [90]. Many solutions have been proposed, ranging from slightly modifying local kernels to encourage crossing low probability regions [73, 114, 85] to constructing entirely new kernels which are mixtures of a local and global component [7, 2, 129]. Despite these advances, there is still no general solution for efficiently sampling complicated high dimensional distributions.

We believe deep learning will play an integral role in developing better general solutions. Deep generative models have demonstrated remarkable success in estimating and approximately sampling complicated, high dimensional distributions, achieving state-of-the-art performance in image/audio/video synthesis, computer graphics, physical/engineering simulations, drug discovery, and other domains [56, 70]. In this vignette we discuss the use of deep generative models to design better proposal distributions for use in MH, both by augmenting existing kernels and by constructing entirely new distributions. Most deep generative models use a neural network (NN) to transform a simple base distribution to closely match a prespecified empirical distribution. The setting of posterior computation via MH introduces two practical problems. First, samples from the target are not available prior to sampling, complicating the process of training the NN. Second, each iteration of MH requires computing the acceptance probability, hence evaluating the proposal density. If the proposal is a simple distribution transformed by a NN, then this requires inverting a NN, which is generally impossible, and computing the Jacobian, which can be numerically intractable in high dimensions.

In this vignette we discuss adaptively tuning normalizing flow (NF) proposals as a means of resolving these challenges. Section 2.1 introduces NFs; Sections 2.2-2.3 cover applications to MH and straightforward generalizations; Section 2.4 discusses exciting future research.

2.1 Introduction to normalizing flows

In this section we provide a brief introduction to NFs and highlight useful properties. One method for generating a flexible class of proposal distributions is to transform a simple D-dimensional random variable Z (e.g., $Z \sim N(0, I_D)$) with a diffeomorphism f parameterized by a NN. Carefully tuning f can result in proposals Y = f(Z) that closely conform to the target. Computing the acceptance probability in each iteration of MH requires evaluating the proposal density,

(1)
$$\pi_Y(y) = \pi_Z(f^{-1}(y))|J_{f^{-1}}(f^{-1}(y))|$$

where π_Z is the density of Z and $J_{f^{-1}}$ is the Jacobian of f^{-1} . Inverting NNs is generally intractable, and evaluating Jacobians is $O(D^3)$ in the worst case.

NFs impose additional structure on f to resolve these problems. Specifically, discrete NFs (DNFs) decompose f as the composition of K simple component functions:

$$(2) f = f_K \circ \cdots \circ f_1.$$

Component functions are constructed to facilitate fast inversion (either exactly or approximately) and fast Jacobian calculations (e.g., by ensuring Jacobians are upper/lower triangular). The change of variables rule becomes

(3)
$$\pi_Y(y) = \pi_Z(f^{-1}(y)) \prod_{i=1}^K |J_{f_i^{-1}}(z_i)|$$

where $f^{-1}=f_1^{-1}\circ\cdots\circ f_K^{-1}$ and $z_i=f_{i+1}^{-1}\circ\cdots\circ f_K^{-1}(y)$ with $z_K=y$. By the inverse function theorem, $J_{f_i^{-1}}=J_{f_i}^{-1}$, so it is sufficient to compute the Jacobian of f_i or f_i^{-1} . For example, a *planar* normalizing flow [137] uses component functions

$$(4) f_i(z) = z + a_i h(w_i^T z + b_i)$$

where $a_i, w_i \in \mathbb{R}^D$, $b_i \in \mathbb{R}$ are parameters to be tuned and h is an invertible, differentiable nonlinearity applied elementwise. The matrix determinant lemma allows one to express the Jacobian as

(5)
$$|J_{f_i}(z)| = 1 + h'(w_i^T z + b) a_i^T w_i$$

which is O(D) to compute. Planar flows are not invertible for all choices of parameters and nonlinearities, however efficient constrained optimization algorithms are available which ensure invertibility [137]. Planar flows have relatively limited expressivity, and many layers may be needed to construct suitably complicated high dimensional proposals. Improved component functions have been proposed, including radial [137], spline [34], coupling [31], autoregressive [69], etc. See [70] for a review of NFs and [122] for theory on the expressively of discrete flows.

Continuous normalizing flows (CNFs) [25] are an extension of the discrete framework, potentially enhancing expressivity while requiring fewer parameters and lower memory complexity. The key insight is to reconceptualize DNFs as a method for computing the path x(t) of a particle at discrete times $t \in \{0, 1/K, 2/K..., 1\}$. The initial location x(0) is drawn from Z. At time 1/K, the location is updated to $x(1/K) = f_1(x(0))$. This is repeated iteratively, moving from x(i/K) at time i/K to $x((i+1)/K) = f_i(x(i/K))$ at time i+1. The result is a path (x(0),...,x(1)) where the final location is a sample from Y. CNFs consider the limit $K \to \infty$, with the intuition that one can obtain a more flexible distribution for Y by flowing samples of Z through continuous paths

instead of discrete paths. This can be formalized as the initial value problem

(6)
$$\frac{dx(t)}{dt} = f(x(t), t)$$

where f is a function parameterized by a NN and x(0) is a sample from Z. In practice equation (6) cannot be solved analytically, however approximate samples of Y can be generated using an ODE solver. Euler's method with a step size of 1/K exactly recovers a DNF with K layers, but greater expressivity can be obtained using higher order solvers. This framework has a number of surprising technical advantages; see [25] for an exposition.

2.2 Normalizing flow proposals

In this section we outline modern methods for constructing proposals with NFs. Throughout, we denote the D-dimensional target density by

(7)
$$\pi(x) \propto \exp(-U(x))$$

with unknown normalizing constant and known potential $U: \mathbb{R}^D \to \mathbb{R}$. A NF with parameters ϕ will be denoted $f_{\phi}: \mathbb{R}^D \to \mathbb{R}^D$; this yields a new density $\hat{\pi}_{\phi}$ by pushing forward a simple random variable Z with density π_Z .

Independent proposals The simplest approach is to use a NF to generate proposals in independent MH [15]. At each iteration, a proposed state x' is generated by pushing a sample of Z through the NF. This state is accepted with probability

(8)
$$\operatorname{acc}(x, x') = \min \left\{ 1, \frac{\pi(x')\hat{\pi}_{\phi}(x)}{\pi(x)\hat{\pi}_{\phi}(x')} \right\}$$

where x is the current state. In high dimensions, almost all choices of ϕ will result in low overlap between $\hat{\pi}_{\phi}$ and π , hence small acceptance ratios and poor mixing. Consequently, we focus our discussion on more elaborate proposals which result in better practical performance.

Dependent proposals A more practical approach is to let proposals depend on the current state. This can be achieved by using a larger NF $f_\phi:\mathbb{R}^D imes \mathbb{R}^M o \mathbb{R}^D imes \mathbb{R}^M$ which maps the current state x and M-dimensional noise z to a proposal x' and transformed noise z'. The M dimensional noise can be thought of as an auxillary parameter such as momentum or temperature in dynamics based MCMC. [147] construct a dependent proposal which is symmetric, thus eliminating the ratio of proposal densities in equation (8) and reducing the problem of extremely low early acceptance rates. The proposal is constructed in two stages: first, sample $u \sim \text{Uniform}[0,1]$ and z from Z. If u > 0.5, propose x' using $(x', z') = f_{\phi}(x, z)$. Otherwise propose x' using $(x', z') = f_{\phi}^{-1}(x, z)$. Using a mixture of f_{ϕ} and f_{ϕ}^{-1} ensures that x' is as likely to be proposed when starting at x as x is to be proposed when starting at x'. Key to the proof of symmetry is the assumption that the NF is volume preserving. This is a restrictive assumption: current volume preserving architectures are outperformed by non-volume preserving architectures.

Mixture kernels Higher initial acceptance rates can be obtained by combining NF proposals with classical kernels, for example by alternating proposing samples with HMC and a conditional flow. Samples from the classical kernel provide data with which to tune the NF. Eventually, the NF becomes a good approximation to the posterior, proposing efficient global moves and resulting in better mixing than the classical kernel alone. [40] construct a proposal which deterministically alternates between approximately 10 MALA proposals for every one independent NF proposal. The resulting sampler efficiently explores multimodal distributions: MALA locally explores each mode, and NF teleports the chain between modes. It is critical to initialize the sampler with at least one particle in each mode, as the local dynamics are unlikely to discover new modes on their own. The algorithm is shown to converge with an exponential rate in the continuous time limit. Partial ergodic theory is available when the flow is adaptively learned by minimizing the KL divergence, although other loss functions remain unstudied.

Augmenting existing kernels The previously discussed mixtures rely on classical kernels for local exploration until there is sufficient data to train the NF. An alternate approach is to use NFs to augment classical kernels - that is, to improve the classical kernel as the chain runs instead of tuning a separate, auxillary kernel. We use HMC as an example, wherein a new state x^\prime is proposed by drawing a momentum $\nu \sim N(0,I_D)$ and approximating the resulting Hamiltonian dynamics (usually) with the leapfrog integrator. One time step of the approximation proceeds by taking a half step of the momentum

(9)
$$\nu_{1/2} = \nu - \frac{\varepsilon}{2} \nabla U(x)$$

where x is the current state and ε is the step size of the integrator. This is used to update the position

$$(10) x' = x + \varepsilon \nu_{1/2}$$

which is then used to update the momentum,

(11)
$$\nu' = \nu - \frac{\varepsilon}{2} \nabla U(x')$$

The process is repeated a prespecified number of times to generate a final proposal; the final momentum is disregarded. The resulting proposal is symmetric and volume preserving, resulting in a simple acceptance ratio. Crossing low-probability regions requires a large velocity, which is unlikely if the momentum is sampled from a Gaussian. [77] use NFs to learn a collection of maps which dynamically rescale the momentum and position to encourage exploration across low probability regions. Specifically, the momentum half step is replaced by

$$\nu_{1/2} = \exp(S_{\nu}(x)) \odot \nu - \frac{\varepsilon}{2} \exp(Q_{\nu}(x)) \odot \nabla U(x) + T_{\nu}(x)$$

where \odot is the elementwise product, S_{ν} is a NF that rescales the momentum, Q_{ν} is a NF that rescales the gradient, and T_{ν} is a NF that translates the momentum. Similarly, the position update is replaced with (13)

$$x' = \exp(S_x(\nu_{1/2})) \odot x + \varepsilon \exp(Q_x(\nu_{1/2})) \odot \nu_{1/2} + T_x(\nu_{1/2})$$

where S_x , Q_x , and T_x are NFs. The momentum is updated again with equation (12) using x' in place of x, and the entire procedure is iterated. When all of these NFs are zero, we exactly recover HMC. Allowing the NFs to be nonzero results in a very flexible family of proposal distributions which can be adaptively tuned to propel the sampler out of low probability regions by rescaling and translating the momentum/position. The invertibility and tractable Jacobians allows efficient calculation of the proposal density. This presentation has been simplified from [77], which also includes random directions, random masking, and conditions NFs on the leapfrog iteration. So far, the above augmentation technique has only been applied to HMC. However there is a broad class of dynamical systems that can be used to generate proposals, including Langevin dynamics, relativistic dynamics, Nose-Hoover thermostats, and others [86]. NFs can be used to augment all of these algorithms using the same recipe as above.

2.3 Tuning proposals

Appropriately tuning NF parameters is critical for good mixing. In practice, tuning is often performed by adaptively minimizing a loss. In this section we cover a variety of candidate loss functions, including measure-theoretic losses, summary statistics, and adversarial approaches.

Statistical deviance The simplest approach is to define a function d measuring how close the proposal is to the target and then to find NF parameters minimizing $d(\hat{\pi}_{\phi}, \pi)$. Let \mathcal{G} be a space of probability densities and $d: \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ be any function measuring the distance/discrepancy/deviance between two probability measures. We assume

- 1. $d(\rho, \rho) = 0$ for all $\rho \in \mathcal{G}$.
- 2. $d(\rho, \rho') > 0$ for $\rho \neq \rho' \in \mathcal{G}$, with equality interpreted as equality almost everywhere.
- 3. $d(\hat{\pi}_{\phi}, \pi)$ has a gradient with respect to ϕ , $\nabla_{\phi} d(\pi_{\phi}, \pi)$.

Conditions (1) and (2) ensure $d(\hat{\pi}_{\phi},\pi)=0$ if and only if $\hat{\pi}_{\phi}=\pi$, hence minimizing d is a reasonable way to approximate the target. Condition (3) allows optimization with gradient based methods. Weaker notions of differentiablility are sufficient, such as having a tractable subgradient.

For example, d may be the forward KL divergence,

(14)
$$\operatorname{D}_{\mathrm{KL}}(\pi \| \hat{\pi}_{\phi}) = \int_{\mathbb{R}^{D}} \pi(x) \log \left(\frac{\pi(x)}{\hat{\pi}_{\phi}(x)} \right) dx$$

Adaptive estimation can be performed by alternating between generating a sample via MH and updating NF parameters using the gradient of (14) [15]. The gradient can

be estimated via Monte Carlo using previous samples. Under technical assumptions on the NF and the target, the resulting Markov chain is ergodic with the correct limiting distribution [15].

Other viable choices for d include the Hellinger distance, the (sliced) Wasserstein distance, the total variation distance, etc. Many of these are as-of-yet unexplored as a means of adaptively estimating flows, and it is unclear which will result in the best performance. The main limitation with approaches in this class is that minimizing a difference only indirectly targets good mixing; in the following we consider directly targeting good mixing with MCMC diagnostics.

Mixing summary statistics A high quality global approximation of the target may not be required for sufficiently good mixing, especially if NFs are used in conjunction with local kernels such as HMC. Using distance based losses in these situations is unnecessarily ambitious and better practical performance may be attained by switching to a loss function which directly targets good mixing. Ideally one would maximize the effective sample size, but this depends on the entire history of the chain and is in general slow to compute. Instead [77] propose minimizing the lag-1 autocorrelation, which is equivalent to maximizing the expected squared jump distance [124]:

(15)
$$\log(\hat{\pi}_{\phi}, \pi) = E[||x - x'||_2^2 \operatorname{acc}(x, x')]$$

where the expectation is over the target and any auxiliary variables used to sample x'. This can be estimated using samples x_i , i = 1, ..., S from the first S iterations of the chain by generating a proposal x'_i starting at each x_i and averaging:

(16)
$$\log(\hat{\pi}_{\phi}, \pi) \approx \frac{1}{S} \sum_{i=1}^{S} ||x_i - x_i'||_2^2 \operatorname{acc}(x_i, x_i').$$

This loss depends on ϕ implicitly through the x_i' . Naively optimizing this loss does not guarantee good mixing across the entire space - for example, the chain may bounce between two distant modes. To solve these problems, [77] add a reciprocal term and instead optimize

(17)
$$\ell_{\lambda}(\hat{\pi}_{\phi}, \pi) = \frac{\lambda}{\log(\hat{\pi}_{\phi}, \pi)} - \frac{\log(\hat{\pi}_{\phi}, \pi)}{\lambda}$$

where $\lambda>0$ is a tuning parameter. The reciprocal term penalizes states where the expected squared jump distance is small. [77] add a term of the same form to encourage faster burn-in. The composite loss is used to train an augmented variant of HMC and results in a sampler which efficiently moves between well-separated modes.

Other summary statistics can be integrated into this framework, possibly considering lag-k autocorrelations or multiple chain summaries such as the Gelman-Rubin statistic [42]. One concern with this class of loss functions

is that no single summary statistic can detect when a chain has mixed, and naively optimizing one statistic may result in pathological behaviour that is hard to detect. In the following we discuss a different strategy which may strike a middle ground between ambitious distance based methods and narrow summary statistic based methods.

Adversarial training Generative adversarial networks (GANs) [47, 51] pit two NNs against each other in a minimax game. The first player is a generator which transforms noise into samples that look like real data; the second player is a discriminator which tries to determine whether an arbitrary sample is synthetic or real. GANs may be applied to MCMC by taking the proposal distribution to be the generator and training a discriminator to distinguish between proposals and previous samples of the target. [147] use this idea to adaptively train a NF proposal which dramatically outperforms HMC on multimodal distributions.

Many improvements are possible by leveraging modern ideas from the GAN literature. Conditional GANs [101] allow the discriminator and generator to condition on external variables. For example, one could construct a tempered adversarial algorithm by conditioning on a temperature variable, possibly accelerating the mixing of annealed MCMC. Complicated GAN structures are prone to mode collapse, hence these generalizations will likely require modified loss functions [152, 93, 64, 163] and regularization [52, 126, 140, 102].

2.4 Future directions

We have introduced several different kernel structures and losses which can be combined to develop new adaptive MCMC algorithms. In this section we discuss shortcomings of the proposed approach, as well as avenues for exciting long-term research.

Theoretical guarantees So far, partial ergodic theory is only available in the simplest case of tuning an independent NF proposal by adaptively minimizing the KL divergence [15]. Dependent/conditional proposals and augmented kernels are not well studied, and no guarantees are available when adaptively minimizing summary statistic or adversarial based losses. This is particularly concerning for summary statistic based losses, as it is not clear that minimizing (e.g., lag-1 autocorrelation) is enough to guarantee ergodic averages converge to the correct values. Precise theoretical results will provide insights into when/why these methods succeed/fail, and are a necessary precursor to widespread adoption of NF sampling.

Constrained posteriors In this vignette we only consider the case where the target is supported over Euclidean space, however in some applications the target is supported over a Riemannian manifold (e.g., the sphere or positive semidefinite matrices). Most manifold sampling algorithms rely on approximating dynamics defined either intrinsically on the manifold or induced by projecting

from ambient space. These dynamics based methods may be inferior to NF kernels for multimodal distributions. Recent work has successfully generalized NFs to Riemannian manifolds, although these constructions typically place significant restrictions on geometry (e.g., diffeomorphic to a cross product of spheres [138]) or rely on high-variance estimates of Jacobian terms [94]. Loss functions measuring the distance between a proposal and the target may be harder to define and compute over manifolds. New architectures for manifold valued NFs and improved estimation techniques could facilitate efficient sampling in a wide class of models with non-Euclidean supports.

Our discussion also neglected to mention discrete parameters. Discrete parameters occur routinely in Bayesian applications, including clustering/discrete mixture models, latent class models, and variable selection. Specific NF architectures have been constructed to handle discrete data [151, 177], but current approaches are relatively inflexible and cannot be made more flexible by naively adding more NN layers, limiting their utility within MH. A more promising direction is to leverage the flexibility of continuous NFs by embedding discrete parameters in Euclidean space and sampling from an augmented posterior. Several variants of HMC have been proposed to accommodate piecewise discontinuous potential functions [121, 103, 32], with recent implementations such as discontinuous HMC (DHMC) [115] achieving excellent practical performance sampling ordinal variables. However, embedding based methods struggle to sample unordered variables - here the embedding order is arbitrary, with most embeddings introducing multimodality in the augmented posterior. NFs have successfully augmented continuous HMC [77] to handle multimodal distributions; the same strategy is promising for improving DHMC.

Automated proposal selection A priori it is unclear which NF architecture, kernel structure, and loss will result in the most efficient mixing for sampling a given posterior. Running many Markov chains with different choices can be time consuming, and a large amount of computational effort may be wasted if some chains mix poorly. Tools for automatic architecture/kernel/loss selection would greatly improve the accessibility of the proposed methodology. This goal is difficult in general given (1) the space of possible samplers is huge, (2) different architectures and kernels are not always comparable, and (3) good mixing is impossible to quantify with a single numerical summary.

Ideas from reinforcement learning, sequential decision making, and control theory could provide principled algorithms for exploring the space of possible samplers. One could define a state space of kernel/loss pairs, $(\hat{\pi}_{\phi}, L)$, which an agent interacts with by running adaptive MCMC. After each action, the agent observes sampler outputs such as trace plots and summary statistics. The goal is to develop a policy for choosing the next kernel/loss pair to

run while maximizing some cumulative reward, such as cumulative effective sample sizes across all chains. As an initial attempt, one could restrict kernels to all have the same structure, such as HMC/NF mixtures where only the NF architecture is changing, and the loss function to be a simple parametric family, such as the lag-1 loss with different tuning parameters. This facilitates a parameterization of the state-space and allows application of existing continuous-armed bandit algorithms [1, 159]. Constructing a sequential decision making algorithm that can efficiently explore kernel/loss pairs with fundamentally different kernel structures and loss functions is an open challenge, which will likely require better understanding of the theoretical relationships between the different proposed kernel structures as well as the dynamics which result from minimizing different types of losses.

We expect broad patterns to emerge with increasing use of NF, with certain architectures/kernels/losses performing consistently well in specific classes of problems. For example, the authors have observed that discrete spline flows work very well for sampling from Gaussian mixture models. These heuristics could be collected in a community reference manual, allowing statisticians to quickly find promising candidate algorithms for their model class, dimension, features of the data, etc. Crowd-sourcing the construction and maintenance of this manual could enable statisticians to stay up-to-date with NFs, despite the rapid pace of ML research.

Accelerated tuning The recipe presented in this vignette is to (1) choose a NF kernel structure, (2) choose a loss, and (3) adaptively estimate parameters starting from a random initialization. Starting from a random initialization in step (3) is inefficient. Transfer/meta learning may provide tools for accelerating tuning by avoiding random initialization. For example, iterative model development and sensitivity analysis often involve repeating the same inferences with slightly different prior specifications. NF parameters estimated for one prior specification could be used to initialize the sampler for other prior specifications, potentially eliminating the need for adaptive tuning.

A more difficult task is handling targets with similar structures, but different dimensions. For example, consider a Bayesian sparse logistic regression model classifying Alzheimer's disease status using vectorized images of brains. Interest is in sampling coefficients β_I from the posterior $\pi(\beta_I \mid A, I)$ where $A = (A_1, ..., A_n)$ is a set of disease indicators and $I = (I_1, ..., I_n)$ is a set of brain images. Perhaps additional covariates for each subject are collected at a later stage, such as gene expression vectors $G = (G_1, ..., G_n)$. Intuitively, there should be strong similarities between the updated posterior $\pi(\beta_I, \beta_G \mid A, I, G)$ and the original posterior $\pi(\beta_I \mid A, I)$, however this is difficult to formalize because the posteriors have different dimensions.

A promising approach is to parameterize the initial sampler in a dimension-free manner, for example by defining a kernel which proposes an update for the ith coefficient depending only on the potential $U(\beta)$, the gradient in that direction $\partial_{\beta_i}U(\beta)$, and auxillary variables in that direction. This kernel can be tuned while sampling $\pi(\beta_I \mid A, I)$ with any of the aforementioned loss functions, and then automatically applied to sample $\pi(\beta_I, \beta_G \mid A, I, G)$. [46] introduce a related idea for stochastic gradient sampling of Bayesian neural networks with different activation functions. The general methodology remains unstudied for exact sampling. The proposed coordinate-wise strategy cannot leverage correlation between pairs of parameters to propose efficient block updates; solutions to this problem constitute ongoing research.

3. BAYESIAN CORESETS

Large-scale datasets—i.e., those where even a single pass over the complete data set is computationally costly—are now commonplace. MCMC typically requires many passes over the full data set; in the setting of large-scale data, this makes inference, iterative model development, tuning, and verification arduous and error-prone. To realize the full benefits of Bayesian methods in important modern applications, we need inference algorithms that handle the scale of modern datasets.

In the past decade, there has been a flurry of work on approximate Bayesian inference methods that are computationally efficient in the large-scale data regime. One class of methods—including variational inference [66, 158, 11] and Laplace approximations [145, 54]—formulates inference as an optimization problem that can be solved via (scalable) stochastic gradient descent [57, 132]. Because the problem is generally nonconvex, these approaches come with little or no realizable guarantees, and tend to be sensitive to initialization, optimization hyperparameters, and stochasticity during optimization. Another class subsampling MCMC [9, 71, 88, 166, 3]; see Quiroz et al. [130] for a recent survey—run a Markov chain whose transitions depend on a subset of data randomly chosen at each iteration. However, speed benefits can be outweighed by drawbacks, as uniformly subsampling at each step causes MCMC to either mix slowly or provide poor approximation [63, 104, 10, 130, 131]. It is possible to circumvent this restriction by design of an effective control variate for the log-likelihood (see Quiroz et al. [130], Nemeth and Fearnhead [108]), but this is in general model-specific.

At its core, the problem of working with large-scale data efficiently is a question of how to exploit *redundancy* in the data. To draw principled conclusions about a large data set based on a small fraction of examples, one must rule out the presence of unique or interesting additional information in the (vast) remainder of unexamined data. One approach incorporates redundancy directly into its

formulation: *Bayesian coresets* [59]. The key idea is to represent the large-scale data by a small, weighted subset. The coreset can then be passed to any standard (automated) inference algorithm, providing posterior inference at a reduced computational cost.

Coresets come with a number of compelling advantages. First, and perhaps most importantly, coresets preserve important model structure. If the original Bayesian posterior distribution exhibits symmetry, weak identifiability, discrete variables, heavy tails, low-dimensional subspace structure, or otherwise, the coreset posterior typically will exhibit that same structure, because it is constructed using the same likelihood and prior as the original model. This makes coresets appealing for use in complex models where, e.g., a Gaussian asymptotic assumption is inappropriate. Second, coresets are composable: coresets for two data sets can often be combined trivially to form a coreset for the union of data sets [37]. This makes coresets naturally applicable to streaming and distributed contexts [19, Section 4.3]. Third, coresets are inference algorithmagnostic, in the sense that once a coreset is built, it can be passed to most downstream inference methods—in particular, exact MCMC methods with guarantees—with enhanced scalability. Finally, coresets tend to come with guarantees relating the size of the coreset to the quality of posterior approximation.

In this vignette, we will cover the basics of Bayesian coresets as well as recent advances in Sections 3.1 and 3.2, and discuss open problems and exciting directions for future work in Section 3.3.

3.1 Introduction to Bayesian coresets

3.1.1 Setup We are given a target probability density $\pi(\theta)$ for $\theta \in \Theta$ that that is comprised of N potentials $(f_n(\theta))_{n=1}^N$ and a base density $\pi_0(\theta)$,

(18)
$$\pi(\theta) = \frac{1}{Z} \exp\left(\sum_{n=1}^{N} f_n(\theta)\right) \pi_0(\theta),$$

where the normalization constant Z is not known. This setup corresponds to a Bayesian statistical model with prior π_0 and i.i.d. data X_n conditioned on θ , where $f_n(\theta) = \log p(X_n|\theta)$. The goal is to compute or approximate expectations under π ; in the Bayesian scenario, π is the posterior distribution.

A key challenge arises in the large N setting. Bayesian posterior computation algorithms tend to become intractable. For example, MCMC typically has computational complexity $\Theta(NT)$ to obtain T draws, since $\sum_n f_n(\theta)$ (and often its gradient) needs to be evaluated at each step. In order to reduce this $\Theta(NT)$ cost, *Bayesian coresets* [59] replace the target with a surrogate density

(19)
$$\pi_w(\theta) = \frac{1}{Z(w)} \exp\left(\sum_{n=1}^N w_n f_n(\theta)\right) \pi_0(\theta),$$

where $w \in \mathbb{R}^N$, $w \geq 0$ are a set of weights, and Z(w) is the new normalizing constant. If w has at most $M \ll N$ nonzeros, the $\Theta(M)$ cost of evaluating $\sum_n w_n f_n(\theta)$ (and its gradient) is a significant improvement upon the original $\Theta(N)$ cost. The goal is then to develop an algorithm for coreset construction—i.e., selecting the weights w—that:

- 1. produces a small coreset with $M \ll N$, so that computation with π_w is efficient;
- 2. produces a high-quality coreset with $\pi_w \approx \pi$, so that draws from π_w are similar to those from π ; and
- 3. runs quickly, so that building the coreset is actually worth the effort for subsequent fast draws from π_w .

These three desiderata are in tension with one another. The smaller a coreset is, the more "compressed" the data set becomes, and hence the worse the approximation $\pi_w \approx \pi$ tends to be. Similarly, the more efficient the construction algorithm is, the less likely we are to find an optimal balance of coreset size and quality with guarantees.

3.1.2 Approaches to coreset construction There are three high-level strategies that have been used in the literature to construct Bayesian coresets.

Subsampling The baseline method is to uniformly randomly pick a subset $\mathcal{I} \subseteq \{1, ..., N\}$ of $|\mathcal{I}| = M$ data points and give each a weight of N/M, i.e.,

(20)
$$w_n = \frac{N}{M}$$
 if $n \in \mathcal{I}$, $w_n = 0$ otherwise,

resulting in the unbiased potential function approximation

(21)
$$\sum_{n=1}^{N} f_n(\theta) \approx N \left(\frac{1}{M} \sum_{m \in \mathcal{I}} f_m(\theta) \right).$$

This method is simple and fast, but typically generates poor posterior approximations. Constructing the subset by selecting data with nonuniform probabilities does not improve results significantly [59]. Empirical and theoretical results hint that in order to maintain a bounded approximation error, the subsampled coreset must grow in size proportional to N, making it a poor candidate for efficient large-scale inference. Coresets therefore generally require more careful optimization.

Sparse regression One can formulate coreset construction as a sparse regression problem [19, 18, 175],

$$w^* = \operatorname*{arg\,min}_{w \in \mathbb{R}^N_+} \left\| \sum_{n=1}^N f_n - \sum_{n=1}^N w_n f_n \right\|^2 \quad \text{s.t.} \quad \|w\|_0 \le M,$$

where $\|\cdot\|$ is some functional (semi)norm, and $\|w\|_0$ is the number of nonzero entries in w. This optimization problem can be solved using iterative greedy optimization strategies that provably, and empirically, provide a significant improvement in coreset quality over subsampling methods [19, 18, 175]. However, this approach requires

the user to design—and tends to be quite sensitive to—the (semi)norm $\|\cdot\|$, and so is not easy to use for the general practitioner. The (semi)norm also typically cannot be evaluated exactly, resulting in the need for Monte Carlo approximations with error that can dominate any improvement from more careful optimization.

Variational inference Current state-of-the-art research formulates the coreset construction problem as variational inference in the family of coresets [17],

(22)
$$w^* = \underset{w \in \mathbb{R}^N_+}{\operatorname{arg\,min}} \operatorname{D}_{\operatorname{KL}} (\pi_w \| \pi) \quad \text{s.t.} \quad \|w\|_0 \le M.$$

Unlike the sparse regression formulation, this optimization problem does not require expert user input. However, it is not straightforward to evaluate the KL objective,

(23)
$$\log Z - \log Z(w) + \sum_{n=1}^{N} (w_n - 1) \int \pi_w(\theta) f_n(\theta) d\theta$$
,

even up to a constant in w. The difficulty arises because Eq. (23) involves both the unknown normalization constant Z(w) and an expectation under π_w , from which we cannot in general obtain exact draws. This is unlike a typical variational inference problem, where the normalization of the variational density is known and obtaining draws is straightforward. Current research on coreset construction is generally focused on addressing these issues; this is an active area of work, and a number of good solutions have been found [17, 92, 60, 105, 23, 91].

3.2 Notable recent advances

The literature on Bayesian coresets is still in its early stages, and the field is developing quickly. We highlight some key recent developments here.

Coreset data point selection Optimization-based coreset construction methods have tended to take a "one-ata-time" greedy selection strategy to building a coreset, thus requiring a slow, difficult to tune inner-outer-loop [19, 17]. Recent work [23, 105, 60] demonstrates coresets can be built without sacrificing quality by first uniformly subsampling the data set to select coreset points, followed by batch optimization of the weights. This is both significantly simpler and faster than past one-at-a-time selection approaches, while providing theoretical guarantees: for models with a strongly log-concave or exponential family likelihood, after subsampling, the KL divergence of the optimally-weighted coreset posterior converges to 0 as $N \to \infty$ as long as the coreset size $M \gtrsim \log N$ [105]. This guarantee does not say anything about whether one can find the optimal weights, but just that selecting coreset data points by subsampling does not limit achievable quality.

Optimizing the KL divergence Given a selection of coreset points, there remains the problem of optimizing the KL objective over the coreset weights w; this is challenging because one cannot obtain exact draws from π_w , or compute its normalization constant. It is possible to use MCMC to draw from π_w , and to circumvent the normalization constant issue by noting that derivatives are available via moments of the potential functions under π_w , e.g.,

(24)

$$\frac{\partial}{\partial w_n} \mathrm{D_{KL}}(\pi_w || \pi) = -\mathrm{Cov}_w \left[f_n(\theta), \sum_{i=1}^N (1 - w_i) f_i(\theta) \right],$$

where Cov_w denotes covariance under π_w [17, 105]. The key difficulty of this approach is that it requires tuning the MCMC method at each optimization iteration, as the weights w (and hence the target π_w) are changing. Second order methods reduce the number of optimization iterations required significantly [105], and hence the challenge posed by needing to tune MCMC.

Another promising approach is to use a surrogate variational family that is parametrized by the coreset weights w but enables tractable draws and exact normalization constant evaluation [23, 60, 91]. For example, Chen, Xu and Campbell [23] propose using a variational surrogate family q_w such that for all w, $q_w \approx \pi_w$, and then optimizing the surrogate objective function

(25)
$$w^* = \operatorname*{arg\,min}_{w} \mathrm{D}_{\mathrm{KL}} \left(q_w \| \pi \right).$$

Chen, Xu and Campbell [23] set q_w to be a normalizing flow based on sparse Hamiltonian dynamics targeting π_w . Concurrent work by Jankowiak and Phan [60] proposes a similar idea, but based on variational annealed importance sampling [141] as opposed to normalizing flows. In either case, the optimization problem is then just a standard KL minimization over parameters w. Manousakas, Ritter and Karaletsos [91], in contrast, propose using a generic variational family q_λ parametrized by some auxiliary parameter λ to take draws, and adds an additional penalty to the optimization objective to tune q_λ to approximate π_w :

(26)
$$w^{\star}, \lambda^{\star} = \operatorname*{arg\,min}_{w \lambda} \mathrm{D}_{\mathrm{KL}} \left(\pi_{w} \| \pi \right) + \mathrm{D}_{\mathrm{KL}} \left(q_{\lambda} \| \pi_{w} \right).$$

The unknown normalization constant on π_w cancels in the two KL divergence terms, and the $D_{KL}(\pi_w \| \pi)$ term is estimated using self-normalized importance sampling based on draws from q_λ (which should be close to π_w , ideally, due to the additional penalty term). Manousakas, Ritter and Karaletsos [91] use a diagonal-covariance Gaussian family for q_λ , and use an inner-outer loop optimization method in which the inner loop optimizes λ to help ensure that q_λ remains close to π_w .

These two approaches are strongly connected. Consider the optimal auxiliary parameter

(27)
$$\lambda^{\star}(w) = \operatorname*{arg\,min}_{\lambda} \mathrm{D}_{\mathrm{KL}}\left(q_{\lambda} \| \pi_{w}\right),$$

and assume that the family q_{λ} is flexible enough such that $q_{\lambda^{\star}(w)} = \pi_w$ for all w. Then the two approaches are equivalent if we define $q_w = q_{\lambda^{\star}(w)}$:

(28)
$$D_{KL}(\pi_w || \pi) + D_{KL}(q_{\lambda}^{\star}(w) || \pi_w) = D_{KL}(q_w || \pi).$$

The advantage of using a generic family q_{λ} is that it is much easier (and more flexible) than being forced to design a family q_w satisfying $q_w \approx \pi_w$. But self-normalized importance sampling is well-known to often work poorly [22] even when the reverse KL divergence is small, and we still need to take draws from π_w once the coreset is built. The approach of directly designing q_w requires more up front effort, but the optimization is well-behaved, and one can obtain i.i.d. draws directly from q_w afterward.

The tradeoff between the three current state-of-the-art approaches—second-order methods with draws from π_w using MCMC [105], direct surrogate variational methods with $q_w \approx \pi_w$ [23], and parametrized surrogate variational methods using $q_\lambda \approx \pi_w$ [91]—has not yet been explored empirically, and is an open direction for future research.

Optimization guarantees Although variational inference in general is nonconvex, the coreset variational inference problem Eq. (22) facilitates guarantees. In particular, Naik, Rousseau and Campbell [105] obtain geometric convergence to a point near the optimal coreset via a quasi-Newton optimization scheme:

$$(29) ||w_k - w_k^{\star}|| \le \eta^k ||w_0 - w_0^{\star}|| + C,$$

where w_k is the k^{th} iterate, and w_k^{\star} is its projection onto a subset of optimal coreset weights (the optimum may not be unique). The constants η and C are related to how good of an approximation the *optimal* coreset is. If the optimal coreset is exact, then $0 < \eta < 1$ and C = 0.

3.3 Open questions and future directions

Recent advances in coreset construction methods and theory have paved the way for a variety of new developments. In this section we highlight important open problems and areas for investigation.

Complex model structure, data, and symmetry The coresets methodology and theory is now starting to coalesce for the basic model setup in Eq. (18) with a finite-dimensional parameter and conditionally i.i.d. data. Many popular models do not fit into this framework, such as certain network models [58], continuous time Markov chains [6], etc. Some of these models involve computational cost that scales poorly in N—e.g., Gaussian process regression with $O(N^3)$ complexity [167]—and would greatly benefit from a summarization approach. Even some models that technically fit in the framework of Eq. (18), such as certain hierarchical models [12], may be better summarized if more of their latent structure is exposed to the coreset construction algorithm.

Moving beyond the conditionally i.i.d. data setup, we advocate thinking about this problem as *model and data* summarization, broadly construed, as opposed to just the specific case of coresets. At an abstract level, Bayesian coresets are just one particular example of how one can construct a computationally inexpensive parametrized variational family π_w that provably contains (a distribution near) the true posterior π . In general, there is no reason this has to be associated with a sparse, weighted subset of data; we could, e.g., summarize networks with subgraphs [118], summarize high-dimensional data with low-dimensional sketches [89], summarize expensive, complicated neural network structures with simpler ones [117], summarize expensive matrices with low-rank randomized approximations [168], etc. The major question to answer is:

What is the natural extension of coresets, or summarization more broadly, to more sophisticated models beyond Eq. (18)? Is there a common underlying principle, or is efficient summarization a problem that must be solved in a case-by-case manner?

We believe that the key to answering these questions is to understand the connections between Bayesian coresets, subsampling, probabilistic symmetries, and sufficiency in statistical models; see, e.g., [29, 74, 119]. Indeed, the fact that Bayesian coresets work at all is a reflection of the fact that one can use a small subset of data potentials as "approximately sufficient statistics," combined with the symmetry of their generating process. Assuming a fruitful connection is made, we expect that current Bayesian coreset construction methods—which are based on subsampling to select a "dictionary" of potentials, followed by optimization to tune the approximation—will serve as a good template in more general models.

Improved surrogates and optimization Early Bayesian coresets literature [59, 18, 17, 19] suffered from the requirement of taking draws from π_w both during and after construction. Sampling during construction poses a particular challenge: if one intends to use MCMC to take draws from π_w , one needs to continually adapt the MCMC kernel to a changing target π_w as the weights w are refined. Recent developments discussed in Section 3.2 suggest that an easier way to approach the problem is to construct a tractable variational family q_w such that $q_w \approx \pi_w$ for all weights w—whether that is a normalizing flow [23], a variational annealed importance distribution [60], or an optimized parametric surrogate [91]—and then to tune the weights w so that $q_w \approx \pi$. The benefit of this approach is the ability to take exact i.i.d. draws and evaluate the density, which circumvents challenges of adaptive in-the-loop MCMC tuning. This leads to the following question:

How should we construct a tractable, summarization-based variational family such that $q_w \approx \pi_w$ for all w?

For methods based on parametric surrogates [91] that set $q_w = q_{\lambda}^{\star}(w)$, where $\lambda^{\star}(w) = \arg\min_{\lambda} D_{\mathrm{KL}}(q_{\lambda} || \pi_w)$,

there are two major avenues for improvement. The first—and more likely achievable—goal is in the optimization of the parametric surrogate. In particular, the methodology currently involves slow inner-loop optimization of the surrogate, as well as potentially high-variance gradient estimates based on self-normalized importance sampling. Handling these two issues would be a major step forward for this approach. The second important area for future work—which may be far more challenging—is to provide theoretical guarantees on the quality of the coreset that is constructed using this method. The primary difficulty is that the surrogate optimization is as hard to analyze as other generic variational inference problems.

For methods based on direct surrogates [60, 23] where $q_w \approx \pi_w$ for all w, there are again two major areas for improvement. First, current methods involve Hamiltonian dynamics, and so are limited in scope to models with multidimensional real-valued variables; future work should extend these methods to models with a wider class of latent variables. The second area is once again to obtain rigorous theoretical guarantees on the quality of the surrogate. This is likely to be much easier than in the general parametric surrogate case above, as q_w is designed to approximate π_w directly, as opposed to just being a stationary point of a nonconvex optimization problem.

Privacy, pseudo-data, and distributed learning Distributed (or federated) learning is a task in which data are stored in separate data centers, and the goal is to perform a global inference given all the data under the constraint that the data are not transmitted between centers. Both exact [27, 21] and approximate [143, 14] methods exist to perform Bayesian inference in this setting. A common additional constraint is that the data within each center are kept private, in some sense, from the other centers.

Coresets provide a potentially very simple solution to the distributed learning problem (both standard and privacypreserving). In particular, coresets are often composable: if one builds subcoresets (independently and without communication) for subsets of a data set, one can combine these trivially to obtain a coreset for the full data set [36]. Coresets have also been extended to the privacy-aware setting, where one either trains pseudopoints with a differentially private scheme [92] or appropriately noises the coreset before sharing [38]. Subsequently, the data centers can share their privatized summaries freely with one another, or with a centralized repository that performs inference. There is some initial work on distributed Bayesian coresets constructed via sparse regression techniques [19, Section 4.3], but this work was done prior to the advent of modern construction methods. Beyond this, there is no study in the literature dedicated to theory and methods for distributed Bayesian coresets, either privacy-preserving or otherwise.

How do we leverage recent advances in coreset construction to efficiently construct differentially-private Bayesian

coresets suitable for distributed learning problems? What theoretical guarantees on communication cost and coreset quality are possible?

Amortized and minimax coreset construction Bayesian coresets are currently constructed in a model-specific manner by minimizing the KL objective in Eq. (22). In situations where multiple models are under consideration—in exploratory analysis or sensitivity analysis, for example—one would need to re-tune the coreset weights for each model under consideration. Given that these re-tuning problems all involve the same data, they should be closely related; but it is currently an open question how to construct multiple related coresets efficiently. In particular:

How generalizable are coresets? Is there a way to construct one optimized coreset that is appropriate for multiple models? Is there a way to amortize the cost of constructing multiple coresets for multiple models?

One potential direction of future work is to formulate a minimax optimization problem that is similar to Eq. (22), but where there is an outer maximization over a set of candidate models. A major question along these lines is whether it is actually possible to summarize a data set with a single coreset of $M \ll N$ data points such that the coreset provides a reasonable approximation for the worst-case model under consideration. Another possible way to tackle the problem is to amortize the cost of multiple coreset construction, in the spirit of inference compilation [75]. Rather than constructing individual coresets, we train a "coreset construction artifact:" a function that takes as input a candidate model and data subsample, and outputs a set of coreset weights. In other words, we learn how to construct coresets. The most likely candidate for such an artifact is a recurrent deep neural network, as is commonly-used in methods like inference compilation. A major question about this direction to consider is in which data analysis scenarios the cost of building such an artifact is worth the subsequent fast generation of coreset weights.

High-dimensional data and models The coresets approach is designed with a focus on large-scale problems in the sense of the number of data points, N. But in practice, modern large-scale problems tend to also involve high-dimensional data and latent model parameters; the dimension may even grow with N. Empirical results have shown that coresets can be effective in problems with 10-100- dimensional data and parameters, while pseudocoresets [92, 91]—which involves summarizing data with synthetic pseudodata points—have been used successfully on larger problems with 60,000- dimensional parameters and 800- dimensional data. But results in this domain are limited, which leads to the following questions for future work:

When do we expect the coresets approach to work with high-dimensional data and high-dimensional model parameters in general? Is there any modification to the (pseudo)coresets approach required to achieve rigorous guarantees in this setting? How does the difficulty of the coreset weight optimization scale in high dimensions?

We begin with a negative (albeit pathological) example. When a large fraction of the potential functions $(f_n)_{n=1}^N$ encode unique information in the posterior, the coresets approach breaks down; it is not possible to maintain a good posterior approximation upon removing potentials. Manousakas et al. [92, Proposition 1] makes this intuition precise with a simple example. In a d-dimensional Gaussian location model with prior $\theta \sim \mathcal{N}(0,I)$, likelihood $\mathcal{N}(\theta,I)$, and data generated via $X_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,I)$, the optimal coreset of any size M < d satisfies

(30)
$$D_{KL}(\pi_{w^*}||\pi) \gtrsim d \text{ as } d \to \infty,$$

with high probability¹. In some sense, this is unsurprising; the Gaussian location model with large d, despite is mathematical simplicity, is a worst-case scenario for data summarization, as one needs at least d potential functions f_n to span a d-dimensional space.

But in practice, high-dimensional data do not typically exhibit this worst-case behaviour; they often instead exhibit some simpler, lower-dimensional structure. Developing (pseudo)coreset methods that take advantage of that structure is a key step needed to make summarization a worthwhile approach in large-scale modern problems. Furthermore, assuming that the coreset size should generally increase with dimension, additional work is needed to understand how the difficulty of the stochastic weight optimization scales. It is worth investigating whether the recently-developed literature on data distillation in deep learning [165] contains any insights applicable to the Bayesian setting.

Improved automation and accessibility Recent advances in research have, for the first time, made coresets a practical approach to efficient Bayesian computation. However, there is still much work to do to make their use possible by nonexperts. First and foremost, there is a need to develop a general, well-engineered code base that interfaces with common probabilistic programming libraries like Stan and Turing [20, 41]. In addition, there is a need for automated methods to (a) select coreset weight optimization tuning parameters, (b) select coreset size, and (c) assess and summarize the quality of the coreset.

Other divergences Currently, all variational coreset construction approaches optimize the reverse Kullback-Leibler divergence. A straightforward direction for future work would be to investigate the effect of using alternative

(31)
$$\frac{X - (d - M)}{\sqrt{2(d - M)}} \xrightarrow{d} \mathcal{N}(0, 1) \qquad d \to \infty.$$

divergences, e.g. the Rényi divergence [80] or χ^2 divergence [30], in Eq. (22). These will all likely pose similar issues with the unknown normalization constant Z(w), but divergences other than the reverse KL may provide coresets with distinct statistical properties.

4. DISTRIBUTED BAYESIAN INFERENCE

Distributed methods for Bayesian inference address the challenges posed by massive data using a divide-andconquer technique. They exploit distributed computing to reduce the time complexity of Monte Carlo algorithms that require multiple sweeps through the data in every iteration. During the last decade, three main groups of distributed methods have been developed for Bayesian inference. The first class of methods is the simplest and has three steps: divide the data into disjoint subsets and store them across multiple machines, run a Monte Carlo algorithm in parallel on all the machines, and combine parameter draws from all the subsets on a central machine. The last step requires one round of communication, so these approaches belong to the class of *one-shot learning* methods [161, 99, 107, 149, 164, 144, 100, 109, 142, 48, 27, 171, 65, 170, 98, 49, 50, 96, 28]. They are based on a key insight that the subset parameter draws provide a noisy approximation of the true posterior distribution and differ mainly in their combination schemes.

The second class of methods relies on distributed extensions of stochastic gradient MCMC [4, 72, 24, 35], which are typically based on stochastic gradient Langevin dynamics (SGLD) [166, 87]. They also split the data into subsets but have several rounds of communication among the machines. In every iteration, they select a subset with a certain probability, draw the parameter using a modified SGLD update, and communicate the parameter draw to the central machine. The high variance of the stochastic gradients and high communication costs have motivated the development of the third set of methods [16, 127]. They are stochastic extensions of global consensus methods for distributed optimization, such as Alternating Direction Method of Multipliers (ADMM) [13, 123]. They divide the data into subsets, store them on machines, and augment the posterior density with auxiliary variables. These variables are conditionally independent given the parameter, and the parameter's marginal distribution reduces to the target under certain limiting assumptions. The former assumption is crucial for drawing the auxiliary variables in parallel, whereas the latter condition ensures asymptotic accuracy. Every iteration consists of synchronous updates where the machines storing the data draw the auxiliary variables and send them to the central machine that uses them to draw the parameter [154, 136, 155, 127, 156].

Distributed Bayesian methods have three main advantages. First, most of them are algorithm-agnostic and are

The result by Manousakas et al. [92] is stated in terms of the inverse CDF of a χ^2 distribution with d-M degrees of freedom. The $\Omega(d)$ lower bound follows directly by noting that $X\sim\chi^2(d-M)$ implies

easily used with any Monte Carlo algorithm. Second, distributed methods come with asymptotic guarantees about their accuracy. Such results show that approximated and target posterior distributions are asymptotically equivalent under mild regularity assumptions. Finally, they are easily extended to handle application-specific constraints, such as clustering of samples in nonparametric models [111] and privacy-preserving federated learning [67].

We cover the basics of distributed Bayesian inference and recent advances in Section 4.1–Section 4.3, and discuss future research directions in Section 4.4.

4.1 One-shot learning

We provide a brief overview of one-shot learning approaches for distributed Bayesian inference. There is a rich variety of such algorithms available in the literature. We start with the most common setup that assumes the observations are conditionally independent given the parameter, leading to a product form for the likelihood. Let $Y_1^N = (Y_1, \dots, Y_N)$ denote the observed data. The model is specified using the distribution \mathbb{P}_{θ} with density $p(y \mid \theta)$ and p-dimensional parameter $\theta \in \Theta \subseteq \mathbb{R}^p$. Assume that Y_1^N are randomly partitioned into K disjoint subsets. Let $Y_{(j)} = \{Y_{(j)1}, \dots, Y_{(j)M}\}$ be the jth subset $(j = 1, \dots, K)$, where we have assumed that all the subset sample sizes equal M for simplicity. The true and subset j likelihoods are $\ell_N(\theta) = \prod_{i=1}^N p(Y_i \mid \theta)$ and $\ell_{jM}(\theta) = \prod_{i=1}^M p(Y_{(j)i} \mid \theta)$. Let Π be a prior distribution on Θ with density $\pi(\theta)$. Then, the posterior density of θ given Y_1^N is $\pi_N(\theta \mid Y_1^N) = \ell_N(\theta)\pi(\theta)/C_N$, where $C_N = \int_{\Theta} \ell_N(\theta)\pi(\theta)d\theta$ and C_N is finite.

Consensus Monte Carlo (CMC) and its generalizations These methods exploit the observation that the full data posterior can be factored as a product of subset posteriors with tempered priors [144]:

$$\pi_N(\theta \mid Y_1^N) = C_N^{-1} \prod_{j=1}^K \{\pi(\theta)\}^{1/K} \ell_{jM}(\theta)$$

(32)
$$\propto \prod_{j=1}^{K} \pi_M(\theta \mid Y_{(j)}) \equiv \prod_{j=1}^{K} \pi_j(\theta).$$

Here $\pi_M(\theta \mid Y_{(j)})$ (or $\pi_j(\theta)$) is the jth subset posterior density of θ computed using likelihood and prior $\ell_{jM}(\theta)$ and $\{\pi(\theta)\}^{1/K}$. Let $\theta_{(j)t}$ be the parameter draws obtained from $\pi_j(\theta)$ using a Monte Carlo algorithm $(j=1,\ldots,K;t=1,\ldots,T)$ and $\hat{\pi}_j(\theta)$ be an estimate of $\pi_j(\theta)$ obtained using $\theta_{(j)t}$ s. Then, $\prod_{j=1}^K \hat{\pi}_j(\theta)$ is proportional to an estimate of $\pi_N(\theta \mid Y_1^N)$. In the special case that $\pi_j(\theta)$ s are Gaussian, then so is $\pi_N(\theta \mid Y_1^N)$ and weighted average of $\theta_{(j)t}$ s correspond to draws from $\pi_N(\theta \mid Y_1^N)$ [144]. More accurate combination algorithms estimate $\pi_j(\theta)$ using kernel density estimation [107], Weierstrass transform [161], random partition trees [164], Gaussian process regression

[109], and normalizing flows [96], where the last two approaches also use importance sampling to select promising $\theta_{(i)t}$ s for better approximation accuracy.

Median and mean posterior distributions These methods combine the subset posterior distributions using their geometric center, such as the median and mean posterior distributions. The main difference between them and CMC-type approaches is the definition of subset posterior densities. Specifically, the *j*th subset posterior density is

(33)
$$\pi_M(\theta \mid Y_{(i)}) = C_M^{-1} \{\ell_{iM}(\theta)\}^K \pi(\theta) \equiv \tilde{\pi}_i(\theta),$$

where $C_M = \int_{\Theta} \{\ell_{jM}(\theta)\}^K \pi(\theta) d\theta$ is assumed to be finite for posterior propriety. The pseudo-likelihood $\{\ell_{iM}(\theta)\}^K$ in (33) is the likelihood of a pseudo sample resulting from replicating every sample in the *i*th subset K times [99]. This pseudo-likelihood ensures the posterior variance of the subset and true posterior densities are calibrated up to $o_P(N^{-1})$ terms [79, 100, 148]. Similar to the previous methods, $\theta_{(i)t}$ s are drawn in parallel from $\tilde{\pi}_j(\theta)$ s using any Monte Carlo algorithm. Let Π_j be the jth subset posterior distribution with density $\tilde{\pi}_i(\theta)$. Then, its empirical approximation supported on the $\theta_{(j)t}$ s is $\hat{\Pi}_j = T^{-1}\sum_{t=1}^T \delta_{\theta_{(j)t}}(\cdot)$, where $\delta_{\theta}(\cdot)$ is the delta measure supported on θ . The median and mean posterior distributions are approximated using empirical measures $\hat{\Pi}^*$ and $\overline{\Pi}$ that are supported on $\theta_{(i)t}$ s. The weights of $\theta_{(i)t}$ s are estimated via optimization such that $\sum_{j=1}^K \mathsf{d}(\hat{\Pi}^*,\hat{\Pi}_j)$ and $\sum_{j=1}^K \mathsf{d}^2(\hat{\overline{\Pi}},\hat{\Pi}_j)$ are minimized, respectively, where d is a metric on probability measures [99, 149]. If θ is one dimensional and d is the 2-Wasserstein distance, then the α th quantile of the mean posterior equals the average of α th quantiles of the K subset posteriors [79].

Mixture of recentered subset posteriors The final combination algorithm uses a K-component mixture of recentered subset posterior densities in (33). Let $\overline{\theta}_{(j)}$ be the mean of $\pi_M(\theta \mid Y_{(j)})$ and $\overline{\theta} = \sum_{j=1}^K \overline{\theta}_{(j)}/K$. Then, the distributed posterior distribution with density

(34)
$$\tilde{\pi}(\theta \mid Y_1^N) = \sum_{j=1}^K \frac{1}{K} \tilde{\pi}_j (\theta - \overline{\theta} + \overline{\theta}_j),$$

approximates $\pi_N(\theta \mid Y_1^N)$, where $\tilde{\pi}_j$ is defined in (33) [171, 170]. To generate draws from $\tilde{\pi}(\theta \mid Y_1^N)$ in (34), we obtain the empirical approximation of the distributed posterior $\tilde{\Pi}$ with density $\tilde{\pi}(\theta \mid Y_1^N)$ as

(35)
$$\hat{\tilde{\Pi}} = \sum_{j=1}^{K} \sum_{l=1}^{T} \frac{1}{KT} \delta_{\hat{\theta} + \theta_{(j)l} - \hat{\theta}_{j}}(\cdot),$$

where $\hat{\theta}_j = \sum_{l=1}^T \theta_{(j)l}/T$ and $\hat{\theta} = \sum_{j=1}^K \hat{\theta}_j/K$. The K-mixture $\hat{\Pi}$ and geometric centers $\hat{\Pi}^*, \hat{\overline{\Pi}}$ are similar in that the atoms of the empirical measures are transformations of

the subset posterior draws. The main difference between them lies in their approach to estimating the weights of the atoms. All the atoms of $\hat{\Pi}$ have equal weights (i.e., $(KT)^{-1}$), whereas the atom weights of $\hat{\Pi}^*$ and $\overline{\Pi}$ are non-uniform and estimated via an optimization algorithm.

Asymptotics The large sample properties of the posterior estimated in one-shot learning, denoted as $\Pi_{D,N}$, are justified via a Bernstein-von Mises (BvM) theorem; however, these results are only known for the last two methods and not for the CMC-type approaches [79, 100]. A BvM for $\Pi_{D,N}$ shows that it is asymptotically normal under mild assumptions as K and N tend to infinity. The center of the limiting distribution is specific to the combination algorithm, but the asymptotic covariance matrix equals I_0/N , where I_0 is the Fisher information matrix computed using $Y \sim \mathbb{P}_{\theta_0}$. This shows that the asymptotic covariance of the true and distributed posteriors are calibrated up to $o_P(N^{-1})$ terms. Under these assumptions,

(36)
$$\|\Pi_{D,N}(\cdot \mid Y_1^N) - \Pi_N(\cdot \mid Y_1^N)\|_{TV} \le \|\tilde{\theta} - \hat{\theta}\|_2$$

ignoring $o_P(N^{-1/2})$ terms, where $\|\cdot\|_{\text{TV}}$ is the total variation distance, $\hat{\theta}$ is the maximum likelihood estimate (MLE) of θ computed using Y_1^N , and $\tilde{\theta}$ is a center of the K subset MLEs of θ : $\hat{\theta}_1,\ldots,\hat{\theta}_K$. They satisfy $\|\hat{\theta}_j-\theta_0\|_2=o_P(M^{-1/2})$, so $\|\tilde{\theta}-\theta_0\|_2=o_P(M^{-1/2})$ because $\tilde{\theta}$ is a center of the subset MLEs. Furthermore, $\|\hat{\theta}-\theta_0\|_2=o_P(N^{-1/2})$ and combining it with the previous result implies that $\|\tilde{\theta}-\hat{\theta}\|_2=o_P(M^{-1/2})$, which does not scale in K. This shows that the bias of $\Pi_{D,N}$ in approximating Π_N does not decrease as K increases, and that K does not generally impact the approximation accuracy of $\Pi_{D,N}$, unless $\tilde{\theta}$ is a root-N consistent estimator of θ_0 .

Notable recent advances One-shot learning, except CMC-type methods, has been generalized to dependent data. In time series data, smaller blocks of consecutive observations form the subsets and preserve the ordering of samples. A measure of dependence, such as the mixing coefficient, dictates the choice of K. The subset pseudo-likelihood in (33) is modified to condition on the immediately preceding time block to model the dependence and raised to a power of K. For one-shot learning in hidden Markov models with mixing coefficient ρ , the distributed posterior estimated using (35) with the modified pseudo-likelihood and $K = o(\rho^{-M})$ satisfies (36) [162]. These results have been generalized to a broader class of models, but guidance on the choice of K remains underexplored [120].

Posterior computations in Gaussian process (GP) regression fail to scale even for moderately large N [133, 8]. One-shot learning has addressed this challenge but with no theoretical results [99, 149]. The choice of K here depends on the smoothness of the regression function.

Assuming a higher order of smoothness of regression functions guarantees accurate estimation on the subsets for larger values of K. Specifically, if the regression function is infinitely smooth, the predictor lies in [0,1], and $K = O(N/\log^2 N)$, then the decay rates of estimation risks for the distributed and true posterior distributions depend only on N and are asymptotically equivalent. In more general problems where the regression function belongs to the Hölder class of functions on $[0,1]^D$ with smoothness index α , the upper bound for K depends on N,D, and α for guaranteeing optimal decay rate of the estimation risk [49]. These results have been generalized to varying coefficients models [50].

Limitations The main limitation of one-shot learning methods is their reliance on the normality of the subset posterior distributions. Scaling of the parameter draws on the subsets helps in some cases but fails to generalize beyond the family of elliptical posterior distributions [146, 157]. [28] identify three additional problems for oneshot learning. First, subset posteriors fail to capture the support of a multimodal posterior with a high probability. Second, a subset posterior can be substantially biased and fail to be a reasonable approximation of the true posterior, violating another major assumption. Finally, subset posterior draws fail to provide information about the tails of the true posterior, resulting in poor estimates of tail event probabilities. A key observation of [28] is that communication among machines is necessary for improving the approximation accuracy of subset posteriors.

4.2 Distributed stochastic gradient MCMC

Langevin Monte Carlo uses the gradient of $\log \pi_N(\theta \mid Y_1^N)$ for generating θ proposals in a Metropolis-Hastings sampling scheme [106]. The gradient computation requires cycling through all the samples, which is prohibitively slow for a large N. SGLD bypasses this problem by subsampling a size n subset S_n of $\{1,\ldots,N\}$ and proposing θ in the (t+1)th iteration given θ_t using a noisy approximation of the gradient $g_N(\theta) = \nabla \log \pi_N(\theta \mid Y_1^N)$ as follows:

(37)
$$\theta_{t+1} = \theta_t + \frac{h_t}{2} \, \hat{g}_n(\theta_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, h_t I),$$
$$\hat{g}_n(\theta) = \nabla \log \pi(\theta) + \frac{N}{n} \sum_{i \in S_n} \nabla \log p(Y_i \mid \theta),$$

where $\hat{g}_n(\theta)$ is a noisy estimate of $g_N(\theta)$ in that

$$\mathbb{E}[\hat{q}_N(\theta)/N] \approx \mathbb{E}[q_N(\theta)/N]$$

for every θ . The step size h_t decreases to 0 such that $\sum_{t=1}^{\infty} h_t = \infty$ and $\sum_{t=1}^{\infty} h_t^2 < \infty$. The discretization error of the Langevin dynamics is negligible as $h_t \to 0$, so the rejection probability of θ_t in the Metropolis-Hastings step approaches 0 [166]. In practice, however, $h_t \propto 1/N$ for better mixing and efficiency [16]. This produces a chain

 $\{\theta_t\}$ that does not have the target as the stationary distribution, but it mimics the true continuous-time Langevin dynamics closely and hence has "approximately" the right target.

The distributed SGLD extension (DSGLD) performs the SGLD update on randomly selected subsets [4]. Let $p=(p_1,\ldots,p_K)$ be a vector of positive probabilities such that $p_1+\ldots+p_K=1$, and p_j is the probability of selecting subset j for the SGLD update. At the (t+1)th iteration, simple distributed SGLD extension selects a subset $j_t \sim$ Categorical(p) and defines

(38)
$$\theta_{t+1} = \theta_t + \frac{h_t}{2} \hat{g}_{mj_t}(\theta_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, h_t I),$$

$$\hat{g}_{mj_t}(\theta) = \nabla \log \pi(\theta) + \frac{M}{p_{j_t} m} \sum_{i \in S} \nabla \log p(Y_{(j_t)i} \mid \theta),$$

where S_m is a size m subsample of $\{1,\ldots,M\}$. The chain $\{\theta_t\}$ jumps to the next worker selected for the SGLD update, and this process continues until convergence. This scheme is undesirable due to communication overload; therefore, DGLD samples θ using (38) multiple times on the selected subset before the chain $\{\theta_t\}$ jumps to a new subset. Additionally, the communication bottlenecks are minimized by selecting "optimal" workers with minimum wait times before the chain $\{\theta_t\}$ jumps; see Section 3.2 in [4].

The efficiency gains in DSGLD come at the cost of loss in asymptotic accuracy. The main reason is that the smaller subset sizes imply that the possible subsample combinations on a subset are much smaller than those obtained using the full data in the standard SGLD update. Better gradient surrogates with smaller variance and higher asymptotic accuracy have been developed [24, 35], but the variance of stochastic gradients increases with N, K, and data heterogeneity, resulting in convergence failures [16].

4.3 Asymptotically exact data augmentation

Asymptotically exact data augmentation (AXDA) generalizes DA using stochastic extensions of global consensus optimization algorithms such as ADMM [136, 154, 155]. AXDA has subset-specific auxiliary variables $z=(z_1,\ldots,z_K)\in\prod_{k=1}^K\mathbb{R}^M$ and tolerance parameter $\rho\in\mathbb{R}_+$, which are similar to "missing data" in DA and tolerance parameter in ADMM. Using the notation in (32), z is chosen such that the augmented density satisfies

(39)
$$\pi_{\rho}(\theta, z_1, \dots, z_K \mid Y_1^N) \propto \pi(\theta) \prod_{k=1}^K \ell_{k,\rho}(\theta, z_k),$$

where $\ell_{k,\rho}(\theta,z_k)=p_k(z_k,Y_{(k)})\kappa_{\rho}(z_k,\theta),\ \kappa_{\rho}$ is a kernel such that $\kappa_{\rho}(\cdot,\theta)$ converges weakly to $\delta_{\theta}(\cdot)$ as $\rho\to 0$, and $p_k(z_k,Y_{(k)})$ is such that $\lim_{\rho\to 0}\int\ell_{k,\rho}(\theta,z_k)\,dz_k=\ell_{kM}(\theta)=\prod_{i=1}^Mp(Y_{(k)i}\mid\theta);$ that is, z plays the role of

missing data and preserves the observed data model as $\rho \to 0$, justifying that AXDA is asymptotically exact.

The advantage of the density in (39) is that the z_k s are conditionally independent given θ . In every iteration, z_k s are drawn in parallel on the machines storing $Y_{(k)}$ s. These draws are communicated to the central machine that uses them to draw θ and generates a Markov chain for θ , whose stationary density equals $\pi_N(\theta \mid Y_1^n)$ under mild assumptions. AXDA has been used for Bayesian inference in generalized linear models and nonparametric regression [136, 155], but proper choices of $p_k(z_k, Y_{(k)})$, ρ , and κ_ρ limit the broader application of AXDA. [156] and [127] develop AXDA using ADMM-type variable splitting and Langevin Monte Carlo algorithms. Like DSGLD, repeated communications among the machines diminish the computation gains from distributed computations.

4.4 Open questions and future directions

This section highlights the limitations of distributed inference methodology, important open problems, and areas for future investigation.

High dimensional and dependent data models A variety of options exist for distributed Bayesian inference in independent data models, but they fail to generalize to high-dimensional models. The literature on distributed methods for inference in high dimensional models is sparsely populated [65]. The development of distributed methods that exploit the low dimensional structure in high dimensional problems is desired.

Most distributed methods assume that the likelihood has a product form; see (32). This assumption fails for many time series and spatial models. There are one-shot learning methods for hidden Markov models [162], but they are inapplicable beyond the family of elliptical posterior distributions. No dependent data extensions are available for DSGLD and AXDA algorithms.

Bias and variance reduction The bias between the true and distributed posterior in one-shot learning fails to decay as K increases. For parametric models, (36) shows that the distributed distribution has a bias of the order $o_P(M^{-1/2})$, which is suboptimal compared to $o_P(N^{-1/2})$ order bias of the true posterior. This means that increasing K has no impact on the accuracy of the distributed posterior. One way to bypass this problem is by centering the distributed posterior at a root-N consistent estimator; see [162]. Addressing this problem is useful for Bayesian federated learning, where one-shot learning is increasingly used due to its simplicity [67]. Similarly, developing gradient surrogates with smaller variances is crucial for Bayesian federated learning using Langevin Monte Carlo.

Asynchronous updates Synchronous updates are crucial for convergence guarantees of DSGLD and Langevin Monte Carlo algorithms based on AXDA; however, synchronous updates become expensive as the number of

subsets increases, resulting in diminishing benefits of distributed computations. Asynchronous updates bypass such problems when the subset sizes are similar, but they imply that the $\{\theta_t\}$ chain is not Markov, which rules out conventional tools for proving convergence guarantees. Asynchronous DSGLD and AXDA extensions have numerous practical benefits. [176] have developed asynchronous DA for variable selection and mixed effects model, but its extension to a broader class of models remains unknown.

Generalized likelihoods Bayesian inference using generalized likelihoods has several advantages, including robustness and targeted inference; however, the current literature relies heavily on exploiting the structure of the hierarchical model. Preliminary results are available about the commonalities between AXDA and approximate Bayesian inference [155]. For broader applications, it is interesting to explore distributed extensions of the *cut* posterior in misspecified models [128] and distributed inference in Bayesian models based on generalized likelihoods.

Applications Distributed Bayesian inference has found applications in federated learning [67]. These methods are ideal for Bayesian analysis of multi-center longitudinal clinical studies because the data cannot be moved to a central location due to privacy concerns. Limited examples of such applications are available; therefore, it is interesting to explore such privacy-preserving extensions of distributed methods.

Automated diagnostics and accessibility Automated application and model diagnostics for distributed methods have received little attention. One-shot learning methods are easily implemented using the parallel R package [150]; however, a similar general purpose software for deploying the distributed algorithms in practice remains to be developed. Addressing these challenges is crucial in facilitating the wide applicability of distributed methods.

5. VARIATIONAL BAYES

Although variational approximations are mentioned in passing within previous sections, in this section we provide a vignette focused specifically on Variational Bayesian (VB) methods, which approximate the posterior distribution by a member in a simpler class of distributions through minimizing the KL divergence. Below, we review some recent developments on theory and computation for variational Bayes and outline future directions.

5.1 Introduction to variational Bayes

We first describe our setup for a *statistical experiment*, defined as a pair of a sample space and a set of distributions on the sample space. For each sample size $n \in \mathbb{N}$, suppose that we observe a \mathcal{X}_n -valued sample $\boldsymbol{X}^{(n)}$, where \mathcal{X}_n is a measurable *sample space* equipped with a reference σ -finite measure μ_n . The sample is modeled with a



Fig 1: An illustration of variational Bayes

distribution $P_{\theta}^{(n)} \in \mathcal{P}(\boldsymbol{X}_n)$ determined by a parameter θ in a measurable parameter space Θ_n .

Let $\Pi(\theta)$ be a prior distribution of θ on Θ_n which often comes with a prior density $\pi(\theta)$. If a collection of distributions $\{\mathsf{P}_\theta:\theta\in\Theta_n\}$ is dominated by some measure μ , then Bayes's rule gives the posterior distribution

$$\Pi(\mathrm{d}\theta|\boldsymbol{X}_n) \propto \underbrace{\frac{\mathrm{d}\mathsf{P}_{\theta}}{\mathrm{d}\mu_n}(\boldsymbol{X}_n)}_{\mathrm{likelihood}} \underbrace{\Pi(\mathrm{d}\theta)}_{\mathrm{prior}}.$$

Variational Bayes (VB) aims to provide an approximation to the posterior distribution $\Pi(\cdot|\boldsymbol{X}_n)$. More specifically, VB turns Bayesian computation into a trackable optimization problem. To do this, one first posits a family of approximate distributions \mathcal{Q} called a *variational family*, which is a set of distributions on Θ . The goal is then to find a member of the variational family that minimizes the KL divergence to the exact posterior $\Pi(\cdot|\boldsymbol{X}_n)$:

(40)
$$\widehat{Q} = \operatorname*{arg\,min}_{Q \in \mathcal{Q}} \mathrm{D}_{\mathrm{KL}} \left(Q \| \Pi(\cdot | \boldsymbol{X}_n) \right)$$

See Figure 1 for a simple graphical illustration. The posterior is approximated with the optimal member \widehat{Q} of the family, which is called a *variational posterior*. Statistical inference is then based on the variational posterior \widehat{Q} .

In solving the optimization problem (40), one first writes $\Pi(d\theta|\mathbf{X}_n)=p_{\theta}(\mathbf{X}_n)\Pi(d\theta)/p(\mathbf{X}_n)$, where $p(\mathbf{X}_n):=\int p_{\theta}(\mathbf{X}_n)\Pi(d\theta)$ is the marginal likelihood of \mathbf{X}_n . The KL-divergence above can be written as

$$D_{\mathrm{KL}}(Q \| \Pi(\cdot | \boldsymbol{X}_n)) = \int \log \left(\frac{p(\boldsymbol{X}_n) Q(\mathrm{d}\theta)}{p_{\theta}(\boldsymbol{X}_n) \Pi(\mathrm{d}\theta)} \right) Q(\mathrm{d}\theta)$$

(41)
$$= \underbrace{-\int \log p_{\theta}(\boldsymbol{X}_n) Q(d\theta) + \mathrm{D}_{\mathrm{KL}}(Q||\Pi)}_{=:\Psi(Q,\Pi,\boldsymbol{X}_n):=-ELBO} + \log p(\boldsymbol{X}_n).$$

In the above, we let

$$\Psi(Q, \Pi, \boldsymbol{X}_n) = -\int \log p_{\theta}(\boldsymbol{X}_n) Q(\mathrm{d}\theta) + \mathrm{D_{KL}}\left(Q \| \Pi\right),$$

which we call the *variational objective function*. This is also the negative of the *evidence lower bound (ELBO)*, where the ELBO is $\int \log p_{\theta}(X_n)Q(\mathrm{d}\theta) - \mathrm{D_{KL}}(Q\|\Pi)$ which provides a lower bound of the 'evidence' or the marginal likelihood $\log p(X_n)$ as seen from (41).

Since $p(X_n)$ is a constant with respect to Q, one has $\stackrel{(42)}{}$

$$\widehat{Q} = \operatorname*{arg\,min}_{Q \in \mathcal{Q}} \Psi(Q, \Pi, \boldsymbol{X}_n) = \operatorname*{arg\,min}_{Q \in \mathcal{Q}} \mathrm{D}_{\mathrm{KL}} \left(Q \| \Pi(\cdot | \boldsymbol{X}_n) \right).$$

Hence, minimizing the KL divergence between the variational family and the exact posterior distribution is equivalent to minimizing the variational objective $\Psi(Q,\Pi,\boldsymbol{X}_n)$ or maximizing the ELBO.

When the variational family has certain simple structure, in particular, the so-called *mean field class*, there are efficient computational algorithms for finding \hat{Q} , based on the well-known *CAVI* (coordinate ascent variational inference) algorithm [66, 169] which guarantees convergence to a local minimizer [11]. Let $\theta = (\theta_1, \dots, \theta_d) \in \Theta$ be a d-dimensional parameter, with d potentially large. The mean-field class imposes posterior independence as:

(43)
$$Q(\theta_1, \dots, \theta_d) = \prod_{j=1}^d Q_j(\theta_j),$$

where Q_j is a distribution for θ_j . By taking the derivative of the ELBO with respect to each of the $Q_j(\theta_j)$, one can arrive at the following coordinate ascent update:

(44)
$$\widehat{Q}_{j}(\theta_{j}) \propto \exp\left(E_{Q_{-j}}\left[\log p(\theta_{j}|\theta_{-j}, \boldsymbol{X}_{n}]\right)\right)$$

where $\theta_{-j}=(\theta_1,\ldots,\theta_{j-1},\theta_{j+1},\ldots,\theta_n)$, and the expectation $E_{Q_{-j}}$ is taken with respect to all variational distributions but that of the jth component. CAVI iteratively updates each coordinate by first initializing $Q_j(\theta_j)$ and then updating the variational distribution of each coordinate conditioned on the others according to (44).

When a statistical model has latent structures such as finite mixture models, topic models and stochastic block models, the dimension of latent variables is typically of the same order as the sample size. The CAVI algorithm is not very efficient for large data sets as it requires sweeping through the whole data set before updating the variational parameters at each iteration. *Stochastic variational inference (SVI)* [57] is a popular alternative in this setting. SVI employs stochastic gradient descent by computing the gradient of the ELBO based on mini batches.

Beyond the mean-field class CAVI and SVI critically depend on the mean-field assumption, with this assumption ruling out posterior dependence across parameters and leading to under-estimation of posterior uncertainty. This motivates more complex variational families, which tend to require tailored algorithms. Black-box VI (BBVI) algorithms [132], including gradient based black-box VI, have emerged as a popular class of such algorithms. [62] propose to utilize stochastic natural gradients within black-box VI to improve efficiency and address the common problem of large variance of gradient estimates.

Amortized VB. In traditional variational inference, parameters need to be optimized for each latent variable, which can be computationally intensive. Amortized VI decreases this cost by building a map from data points to the VB family. This map is typically modeled by a deep neural network trained on a data subset. The local VB parameter for the latent variable is computed using the output of the DNN map; this is "amortized" since past computation is used to simplify future computation. Let $f_{\eta}: \mathcal{X} \to \Phi$ be a feedforward neural network with parameters η from the observation space \mathcal{X} to the parameter space Φ of the variational family. For observation x_i , the corresponding latent variable θ_i has conditional distribution $q_{f_{\eta}(x_i)}(\theta_i)$. Amortized variational Bayes finds η through:

(45)
$$\eta^* = \arg\min_{\eta} \mathrm{D_{KL}} \left(\prod_{i=1}^n Q_{f_{\eta}(x_i)}(\theta_i) \| \Pi(\theta_i \mid \boldsymbol{X}_n) \right).$$

Although amortized VI is a general framework, the most popular application is the *variational auto-encoder* (*VAE*). The target generative model for data X is $X = G(Z) + \epsilon$, with Z latent data having a known distribution, ϵ an additive noise independent of Z, and G parametrized by a deep neural network. VAEs are a popular alternative to GANs for training deep generative models. In a VAE, there is an encoder network where the distribution $\Pi(Z \mid X_n, \theta)$ is amortized by a neural network mapping the data points to the variational family.

5.2 Theory of variational Bayes

In order to verify the frequentist optimality properties of Bayesian posteriors, it is common to study contraction rates, model selection consistency, and asymptotic normality (known as Bernstein von-Mises (BvM) theorems). Under the variational Bayes framework, statistical inference is based on the variational posterior instead of the original posterior, so it is natural to study frequentist optimality of VB posteriors.

In the asymptotic regime, we assume data $\boldsymbol{X}^{(n)}$ are generated from $\mathsf{P}_{\theta^\star}^{(n)}$ and $n\to\infty$. The variational posterior

$$\widehat{Q}_n \in \operatorname*{arg\,min}_{Q \in \mathcal{Q}} \Psi(Q, \Pi, \boldsymbol{X}^{(n)}),$$

is said to have the contraction rate ϵ_n if

(46)
$$\mathsf{E}_{\theta^{\star}}^{(n)}[\widehat{Q}_{n}(d(\theta,\theta^{\star}) \leq A_{n}\epsilon_{n}] \to 1$$

as $n \to \infty$ for any diverging sequence $A_n \to \infty$. If the contraction rate ϵ_n matches the *minimax optimal rate*, we say that the variational posterior distribution is optimal.

Recent work [5, 174, 173] provided theoretical conditions under which the variational posterior is optimal. These conditions imply that when the model is appropriately complex and the prior is sufficiently diffuse, which

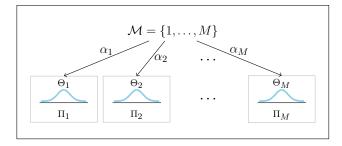


Fig 2: The hierarchical prior distribution

are standard conditions for establishing posterior contraction rates for the original posterior [43], then together with an assumption on the variational gap, the variational posterior distribution also has optimal contraction rates. The variational gap condition assumes there is $Q \in \mathcal{Q}$ such that

(47)
$$\int \mathrm{D_{KL}}(\mathsf{P}_{\theta}^{(n)} \| \mathsf{P}_{\theta^{\star}}^{(n)}) Q(\mathsf{d}\theta) + \mathrm{D_{KL}}(Q \| \Pi) \lesssim n\epsilon_n^2.$$

The left side of (47) is an upper bound on the variational gap $D_{KL}(\hat{Q} || \Pi(\theta | \mathbf{X}_n))$. This upper bound is verified by ensuring that each term on the left is of order $O(n\epsilon_n^2)$. [5] formulate this variational gap condition as an extension of prior mass conditions. If one restricts the VB family to be in the same class as the prior and the parameters to lie in a neighborhood of the true parameter, this condition reduces to the standard prior mass condition.

In addition, [125] and [173] developed variational Bayes theoretic frameworks that can deal with latent variable models. [5] investigated the contraction properties of variational fractional posteriors with the likelihood raised to a fractional power. There are several studies that derived contraction rates of variational posteriors for specific statistical models - for example, mixture models [26], sparse (Gaussian) linear regression [135, 172], sparse logistic linear regression [134], and sparse factor models [113].

5.3 Adaptive Variational Bayes

A notable recent development is a novel and general variational framework for adaptive statistical inference on a collection of model spaces [116]. The framework yields an *adaptive variational posterior* that has optimal theoretical properties in terms of posterior contraction and model selection while enjoying tractable computation.

In general, when performing statistical inference the "regularity" of the true parameter is unknown and adaptive inference aims to construct estimation procedures that are optimal with respect to the unknown true regularity. To do this, one typically prepares *multiple models* with different complexities, e.g. sparse linear regression models with different sparsity, neural networks with different numbers of neurons or mixture models with different numbers of components, and then selects among them. To

achieve adaptivity, frequentists usually conduct (fully data-dependent) model selection before parameter estimation: e.g., via cross-validation or penalization. There is some work on *Bayesian adaptation* by imposing hierarchical priors on a collection of model spaces [44].

Let \mathcal{M} denote a set of model indices and $\{\Theta_m\}_{m\in\mathcal{M}}$ multiple disjoint (sub-)models with different complexities. Let $\Theta_{\mathcal{M}} := \cup_{m\in\mathcal{M}} \Theta_m$ be an encompassing model. A (hierarchical) prior (illustrated by Figure 2) is given as

$$\Pi = \sum_{m \in \mathcal{M}} \alpha_m \Pi_m,$$

where α_m is the prior probability of model Θ_m , $\sum_{m \in \mathcal{M}} \alpha_m = 1$, and Π_m is the prior distribution of θ within model Θ_m . The posterior distribution of $\Theta_{\mathcal{M}}$ is

(48)
$$\Pi(\cdot|\boldsymbol{X}_n) = \sum_{m \in \mathcal{M}} \widehat{\alpha}_m \Pi(\cdot|\theta \in \Theta_m, \boldsymbol{X}_n)$$

where $\widehat{\alpha}_m = \Pi(\theta \in \Theta_m | \boldsymbol{X}_n)$, which can be understood as a weighted average of the posteriors $\Pi(\cdot | \theta \in \Theta_m, \boldsymbol{X}_n)$ on the individual models $(\Theta_m)_{m \in \mathcal{M}}$. If the prior model probabilities $(\alpha_m)_{m \in \mathcal{M}}$ are appropriately chosen, the posterior distribution on the encompassing model can be adaptively optimal [76, 44, 55]. However, computing the posterior of $\Theta_{\mathcal{M}}$ is challenging due to varying "dimensions" of the models and the need to evaluate marginal likelihoods.

[116] address these challenges via variational Bayes adaptation. They approximate posterior (48) using a variational Bayes family over the encompassing model parameter space, using disjoint variational families $\{Q_m\}_{m \in \mathcal{M}}$ over individual models with $Q_m \subset \mathcal{P}(\Theta_m)$:

$$Q_{\mathcal{M}} := \left\{ \sum_{m \in \mathcal{M}} \gamma_m Q_m \mid Q_m \in Q_m \right\}.$$

They show that the variational posterior

$$\widehat{Q}_n \in \operatorname*{arg\,min}_{Q \in \mathcal{Q}_M} \Psi(Q, \Pi, \boldsymbol{X}^{(n)})$$

is of the form

(49)
$$\widehat{Q}_n = \sum_{m \in \mathcal{M}} \widehat{\gamma}_{n,m} \widehat{Q}_{n,m} \in \mathcal{Q}_{\mathcal{M}}$$

for some 'mixing weight' $(\widehat{\gamma}_{n,m})_{m\in\mathcal{M}}$ and 'mixture components' $\widehat{Q}_{n,m}\in\mathcal{Q}_m$ for $m\in\mathcal{M}$. The adaptive variational Bayes framework is summarized in Algorithm 1.

Computation of the adaptive variational posterior reduces to computing variational approximations for each individual model. The framework is general and can be applied for adaptive inference in many statistical models where multiple submodels of different complexities are available. The adaptive variational posterior has optimal contraction rates and strong model selection consistency when the true model is in \mathcal{M} . This theory has been applied to show optimal contraction for a rich variety of models, including finite mixtures, sparse factor models, deep neural networks and stochastic block models.

Algorithm 1 Adaptive variational Bayes

$$\frac{\text{Input: data } X^{(n)}, \text{ prior } \Pi = \sum_{m \in \mathcal{M}} \alpha_m \Pi_m, \text{ variational families } \{\mathcal{Q}_m\}_{m \in \mathcal{M}}.$$

 For every m ∈ M, compute the variational posterior of the submodel Θ_m:

(50)
$$\widehat{Q}_{n,m} \in \arg\min_{Q \in \mathcal{Q}_m} \Psi(Q, \Pi_m, X^{(n)}).$$

• Compute the "optimal model weight" as

(51)
$$\widehat{\gamma}_{n,m} \propto \underbrace{\alpha_m}_{\text{prior}} \times \underbrace{\exp(-\Psi(\widehat{Q}_{n,m}, \Pi_m, \boldsymbol{X}^{(n)}))}_{\text{goodness of fit of } \widehat{Q}_{n,m}}$$

for $m \in \mathcal{M}$

Return: The adaptive variational posterior

(52)
$$\widehat{Q}_n = \sum_{m \in \mathcal{M}} \widehat{\gamma}_{n,m} \widehat{Q}_{n,m}.$$

5.4 Open questions and future directions

Uncertainty quantification of the VB posterior It is well-known that variational posteriors tend to underestimate uncertainty of the posterior, so a central open question is how one can construct computationally efficient VB posteriors producing (a) credible balls with valid frequentist coverage and/or (b) posterior covariance matching that of the true posterior.

There is limited work on theory for statistical inference using the variational posterior, including credible intervals and hypothesis testing. For this, we need theorems to reveal a limiting distribution of the variational posterior as the sample size goes to infinity, just as the Bernsteinvon Mises (BvM) theorem guarantees that the original posterior distribution converges to a Gaussian distribution under certain regularity conditions. An initial promising result along these lines is [160], but there is substantial need for new research for broad classes of models and corresponding variational families.

Theoretical guarantees of gradient-based algorithms Existing theoretical guarantees for VB only apply to the global solution of the variational optimization problem. In practice, this optimization problem tends to be highly nonconvex and algorithms are only guaranteed to converge to local optima. For certain variational families and model classes, these local optima can be dramatically different, so that there is a large sensitivity to the starting point of the algorithm. It is of critical importance to obtain guarantees on the algorithms being used and not just on inaccessible global optima. For example, can one obtain general theoretical guarantees for gradient-based black-box variational inference with or without warm-start conditions?

There is a parallel and growing literature on nonconvex optimization in other contexts, including providing reassurance that local optima can be sufficiently close in some cases [95, 39, 83, 78, 112]. However, to the best of our knowledge, there is no such work on theoretical aspects of local optima produced by variational Bayes.

VB based on generative models Richer variational families can be constructed using deep generative models such as normalizing flows [137, 81]. Due to their impressive flexibility, the resulting variational posterior can approximate a very wide class of target posteriors accurately. Despite its practical usefulness and strong empirical performance, there is no theoretical support for such approaches - for example, providing upper bounds on the variational approximation gap or concentration properties. Choosing the neural network architecture and algorithmic tuning parameters involved in training to maximize computational efficiency and accuracy of posterior approximation is an additional important related area that may benefit from better theoretical understanding.

Online variational inference Given a prior distribution on an unknown parameter, the posterior distribution can be understood as an updated belief after observing the data. The updated posterior distribution can be used as a new prior distribution when new data arrive. The process can be repeated many times for analyzing streaming data [97, 45, 68, 61]. At each step, the VB posterior can be used as a new prior instead of the original one for computational convenience [82, 84, 110]. It would be intriguing to investigate the statistical properties of the sequentially updated VB posterior.

6. DISCUSSION

Tools for Bayesian computation are evolving at a rapid pace, thanks largely to recent developments in machine learning. We highlighted this phenomenon with four vignettes. The first vignette discussed sampling with the aid of generative models, particularly normalizing flows. The next two vignettes discussed different methods for handling the large N regime. Coresets take a variational approach to data compression, with recent methods leveraging deep neural networks to build flexible surrogate families; federated Bayesian learning methods instead distribute posterior computation over many computers. Finally, we covered variational inference, which replaces the posterior with a tractable approximation. Many more vignettes could be written on similar topics, such as accelerating sampling with diffusion based generative models or accelerating approximate Bayesian computation using deep neural networks for data compression. We close with three themes, applicable to all vignettes, that we believe should receive future attention: accelerating inference using previous calculations, improving accessibility with new software, and providing theoretical support for empirically promising algorithms.

The status-quo in Bayesian computation is to start from scratch in each posterior inference problem, such as recomputing coresets after changing the prior, or estimating a new variaitonal approximation when applying an old model to new data. This is inefficient, as posterior inference in similar models must be somewhat informative about posterior inference in the current model. If the two models under consideration are directly comparable, such as posteriors under slightly different priors, then it may be easy to leverage previous calculations, e.g., by using warm starts in optimization routines. Problems arise when the two models have different dimensions, such a hierarchical models with an extra layer of parameters. We are hopeful that methods for similar problems in machine learning – particularly transfer learning – will play a role in developing general solutions for Bayesians.

Another common theme was the need for improved automation and accessibility. Implementing methods involving neural networks or other machine learning techniques in a robust and reliable fashion is a nontrivial task, often requiring significant time and expert knowledge. Given the breakneck speed at which machine learning progresses, careful implementations can be outdated before they have a chance for widespread adoption. The focus should be on developing software which is modular enough to withstand the next machine learning revolution, as well as user-friendly enough to be applied en-masse.

Finally, statisticians should be cautious with wholesale adoption of methods that achieve excellent practical performance at the expense of theoretical guarantees. Fast "approximations" to posterior distributions that can be arbitrarily far from the exact posterior may be useful for black box prediction but fall far short of what is needed for reliable and reproducible Bayesian inferences. This is particularly key in scientific and policy applications in which one needs to appropriately characterize uncertainty in learning from data, acknowledging complexities that arise in practice such as model uncertainty, data contamination etc. Guarantees are necessary to avoid highly misleading inferences and potentially catastrophic conclusions from the types of large and complex datasets that are being generated routinely in the sciences.

REFERENCES

- [1] AGRAWAL, R. (1995). The continuum-armed bandit problem. SIAM Journal on Control and Optimization.
- [2] AHN, S., CHEN, Y. and WELLING, M. (2013). Distributed and adaptive darting Monte Carlo through regenerations. *Artificial Intelligence and Statistics*.
- [3] AHN, S., KORATTIKARA, A. and WELLING, M. (2012). Bayesian posterior sampling via stochastic gradient Fisher scoring. *International Conference on Machine Learning*.
- [4] AHN, S., SHAHBABA, B. and WELLING, M. (2014). Distributed stochastic gradient MCMC. *International Conference on Machine Learning*.
- [5] ALQUIER, P. and RIDGWAY, J. (2020). Concentration of tempered posteriors and of their variational approximations. *The Annals of Statistics*.
- [6] ANDERSON, W. (1991). Continuous-time Markov Chains: an Applications Oriented Approach. Springer.

- [7] ANDRICIOAEI, I., STRAUB, J. E. and VOTER, A. F. (2001). Smart darting Monte Carlo. *The Journal of Chemical Physics*.
- [8] BANERJEE, S., CARLIN, B. P. and GELFAND, A. E. (2014). Hierarchical Modeling and Analysis for Spatial Data. CRC Press, Boca Raton, FL.
- [9] BARDENET, R., DOUCET, A. and HOLMES, C. (2017). On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*.
- [10] BETANCOURT, M. (2015). The fundamental incompatibility of Hamiltonian Monte Carlo and data subsampling. *International Conference on Machine Learning*.
- [11] BLEI, D., KUCUKELBIR, A. and MCAULIFFE, J. (2017). Variational inference: a review for statisticians. *Journal of the American Statistical Association*.
- [12] BLEI, D., GRIFFITHS, T., JORDAN, M. and TENENBAUM, J. (2003). Hierarchical topic models and the nested Chinese restaurant process. Advances in Neural Information Processing Systems.
- [13] BOYD, S., PARIKH, N., CHU, E., PELEATO, B., ECKSTEIN, J. et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends*® *in Machine learning*.
- [14] BRODERICK, T., BOYD, N., WIBISONO, A., WILSON, A. and JORDAN, M. (2013). Streaming variational Bayes. *Advances in Neural Information Processing Systems*.
- [15] Brofos, J., Gabrié, M., Brubaker, M. A. and Leder-Man, R. R. (2022). Adaptation of the independent Metropolis-Hastings sampler with normalizing flow proposals. *International Conference on Artificial Intelligence and Statistics*.
- [16] BROSSE, N., DURMUS, A. and MOULINES, E. (2018). The promises and pitfalls of stochastic gradient Langevin dynamics. Advances in Neural Information Processing Systems.
- [17] CAMPBELL, T. and BERONOV, B. (2019). Sparse Variational Inference: Bayesian Coresets from Scratch. *Advances in Neural Information Processing Systems*.
- [18] CAMPBELL, T. and BRODERICK, T. (2018). Bayesian Coreset Construction via Greedy Iterative Geodesic Ascent. *Interna*tional Conference on Machine Learning.
- [19] CAMPBELL, T. and BRODERICK, T. (2019). Automated Scalable Bayesian Inference via Hilbert Coresets. *Journal of Machine Learning Research*.
- [20] CARPENTER, B., GELMAN, A., HOFFMAN, M., LEE, D., GOODRICH, B., BETANCOURT, M., BRUBAKER, M., GUO, J., LI, P. and RIDDELL, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*.
- [21] CHAN, R., POLLOCK, M., JOHANSEN, A. and ROBERTS, G. (2021). Divide-and-conquer Monte Carlo fusion. arXiv:2110.07265.
- [22] CHATTERJEE, S. and DIACONIS, P. (2018). The sample size required in importance sampling. *Annals of Applied Probability*.
- [23] CHEN, N., XU, Z. and CAMPBELL, T. (2022). Bayesian inference via sparse Hamiltonian flows. *Advances in Neural Information Processing Systems*.
- [24] CHEN, C., DING, N., LI, C., ZHANG, Y. and CARIN, L. (2016). Stochastic gradient MCMC with stale gradients. Advances in Neural Information Processing Systems.
- [25] CHEN, R. T., RUBANOVA, Y., BETTENCOURT, J. and DUVENAUD, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information ProcessingSsystems*.
- [26] CHÉRIEF-ABDELLATIF, B.-E. and ALQUIER, P. (2018). Consistency of variational Bayes inference for estimation and model selection in mixtures. *Electronic Journal of Statistics*.
- [27] DAI, H., POLLOCK, M. and ROBERTS, G. (2019). Monte Carlo fusion. *Journal of Applied Probability*.

- [28] DE SOUZA, D. A., MESQUITA, D., KASKI, S. and ACERBI, L. (2022). Parallel MCMC without embarrassing failures. *International Conference on Artificial Intelligence and Statistics*.
- [29] DIACONIS, P. (1988). Sufficiency as statistical symmetry. *Proceedings of the AMS Centennial Symposium*.
- [30] DIENG, A., TRAN, D., RANGANATH, R., PAISLEY, J. and BLEI, D. (2017). Variational inference via χ -upper bound minimization. *Advances in Neural Information Processing Systems*.
- [31] DINH, L., SOHL-DICKSTEIN, J. and BENGIO, S. (2016). Density estimation using real NVP. *arXiv* preprint *arXiv*:1605.08803.
- [32] DINH, V., BILGE, A., ZHANG, C. and MATSEN IV, F. A. (2017). Probabilistic path Hamiltonian Monte Carlo. *International Conference on Machine Learning*.
- [33] DUNSON, D. B. and JOHNDROW, J. E. (2020). The Hastings algorithm at fifty. *Biometrika*.
- [34] DURKAN, C., BEKASOV, A., MURRAY, I. and PAPAMAKAR-IOS, G. (2019). Neural spline flows. *Advances in Neural Information Processing Systems*.
- [35] EL MEKKAOUI, K., MESQUITA, D., BLOMSTEDT, P. and KASKI, S. (2021). Federated stochastic gradient Langevin dynamics. *Uncertainty in Artificial Intelligence*.
- [36] FELDMAN, D. (2020). Introduction to Core-sets: an updated survey. arXiv:2011.09384.
- [37] FELDMAN, D. and LANGBERG, M. (2011). A unified framework for approximating and clustering data. *Symposium on Theory of Computing*.
- [38] FELDMAN, D., FIAT, A., KAPLAN, H. and NISSIM, K. (2009). Private Coresets. *ACM Symposium on Theory of Computing*.
- [39] FOSTER, D. J., SEKHARI, A. and SRIDHARAN, K. (2018). Uniform convergence of gradients for non-convex learning and optimization. Advances in Neural Information Processing Systems.
- [40] GABRIÉ, M., ROTSKOFF, G. M. and VANDEN-EIJNDEN, E. (2022). Adaptive Monte Carlo augmented with normalizing flows. Proceedings of the National Academy of Sciences.
- [41] GE, H., XU, K. and GHAHRAMANI, Z. (2018). Turing: a language for flexible probabilistic inference. *Artificial Intelligence and Statistics*.
- [42] GELMAN, A. and RUBIN, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*.
- [43] GHOSAL, S., GHOSH, J. K. and VAN DER VAART, A. W. (2000). Convergence rates of posterior distributions. *The Annals of Statistics*.
- [44] GHOSAL, S., LEMBER, J. and VAN DER VAART, A. (2008). Nonparametric Bayesian model selection and averaging. *Electronic Journal of Statistics*.
- [45] GHOSH, S., DELLE FAVE, F. and YEDIDIA, J. (2016). Assumed density filtering methods for learning Bayesian neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [46] GONG, W., LI, Y. and HERNÁNDEZ-LOBATO, J. M. (2018). Meta-learning for stochastic gradient MCMC. arXiv preprint arXiv:1806.04522.
- [47] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. and BENGIO, Y. (2020). Generative adversarial networks. *Communications of the ACM*.
- [48] GUHANIYOGI, R. and BANERJEE, S. (2018). Meta-kriging: Scalable Bayesian modeling and inference for massive spatial datasets. *Technometrics*.
- [49] GUHANIYOGI, R., LI, C., SAVITSKY, T. D. and SRIVAS-TAVA, S. (2022a). Distributed Bayesian inference in massive spatial data. *Statistical Science*.

- [50] GUHANIYOGI, R., LI, C., SAVITSKY, T. D. and SRIVAS-TAVA, S. (2022b). Distributed Bayesian varying coefficient modeling using a Gaussian process prior. *Journal of Machine Learn*ing Research.
- [51] GUI, J., SUN, Z., WEN, Y., TAO, D. and YE, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*.
- [52] GULRAJANI, I., AHMED, F., ARJOVSKY, M., DUMOULIN, V. and COURVILLE, A. C. (2017). Improved training of Wasserstein GANs. Advances in Neural Information Processing Systems.
- [53] HAARIO, H., SAKSMAN, E. and TAMMINEN, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*.
- [54] HALL, P., PHAM, T., WAND, M. and WANG, S. S. J. (2011). Asymptotic normality and valid inference for Gaussian variational approximation. *The Annals of Statistics*.
- [55] HAN, Q. (2021). Oracle posterior contraction rates under hierarchical priors. *Electronic Journal of Statistics*.
- [56] HARSHVARDHAN, G., GOURISARIA, M. K., PANDEY, M. and RAUTARAY, S. S. (2020). A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*.
- [57] HOFFMAN, M. D., BLEI, D. M., WANG, C. and PAISLEY, J. (2013). Stochastic Variational Inference. *Journal of Machine Learning Research*.
- [58] HOLLAND, P., LASKEY, K. and LEINHARDT, S. (1983). Stochastic blockmodels: first steps. Social Networks.
- [59] HUGGINS, J., CAMPBELL, T. and BRODERICK, T. (2016). Coresets for Scalable Bayesian Logistic Regression. Advances in Neural Information Processing Systems.
- [60] JANKOWIAK, M. and PHAN, D. (2021). Surrogate likelihoods for variational annealed importance sampling. *International Con*ference on Machine Learning.
- [61] JEONG, K., CHAE, M. and KIM, Y. Online learning for the Dirichlet process mixture model via weakly conjugate approximation. *Computational Statistics & Data Analysis*.
- [62] JI, G., SUJONO, D. and SUDDERTH, E. B. (2021). Marginalized Stochastic Natural Gradients for Black-Box Variational Inference. Proceedings of the 38th International Conference on Machine Learning.
- [63] JOHNDROW, J., PILLAI, N. and SMITH, A. (2020). No free lunch for approximate MCMC. *arXiv:2010.12514*.
- [64] JOLICOEUR-MARTINEAU, A. (2018). The relativistic discriminator: A key element missing from standard GAN. *arXiv preprint arXiv:1807.00734*.
- [65] JORDAN, M. I., LEE, J. D. and YANG, Y. (2019). Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*.
- [66] JORDAN, M., GHAHRAMANI, Z., JAAKKOLA, T. and SAUL, L. (1999). An introduction to variational methods for graphical models. *Machine Learning* 183–233.
- [67] KIDD, B., WANG, K., XU, Y. and NI, Y. (2022). Federated learning for sparse Bayesian models with applications to electronic health records and genomics. *Pacific Symposium on Bio-Computing*.
- [68] KIM, Y., CHAE, M., JEONG, K., KANG, B. and CHUNG, H. (2016). An Online Gibbs Sampler Algorithm for Hierarchical Dirichlet Processes Prior. *Machine Learning and Knowledge Discovery in Databases*.
- [69] KINGMA, D. P., SALIMANS, T., JOZEFOWICZ, R., CHEN, X., SUTSKEVER, I. and WELLING, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*.

- [70] KOBYZEV, I., PRINCE, S. J. and BRUBAKER, M. A. (2020). Normalizing flows: An introduction and review of current methods. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [71] KORATTIKARA, A., CHEN, Y. and WELLING, M. (2014a). Austerity in MCMC land: cutting the Metropolis-Hastings budget. *International Conference on Machine Learning*.
- [72] KORATTIKARA, A., CHEN, Y. and WELLING, M. (2014b). Austerity in MCMC land: Cutting the Metropolis-Hastings budget. *International Conference on Machine Learning*.
- [73] LAN, S., STREETS, J. and SHAHBABA, B. (2014). Wormhole Hamiltonian Monte Carlo. Proceedings of the AAAI Conference on Artificial Intelligence.
- [74] LAURITZEN, S. (1988). Extremal Families and Systems of Sufficient statistics. Springer-Verlag.
- [75] LE, T. A., BAYDIN, A. G. and WOOD, F. (2017). Inference compilation and universal probabilistic programming. *Artificial Intelligence and Statistics*.
- [76] LEMBER, J., VAN DER VAART, A. et al. (2007). On universal Bayesian adaptation. *Statistics and Decisions-International Journal Stochastic Methods and Models*.
- [77] LEVY, D., HOFFMAN, M. D. and SOHL-DICKSTEIN, J. (2017). Generalizing Hamiltonian Monte Carlo with neural networks. *arXiv preprint arXiv:1711.09268*.
- [78] LI, J., LUO, X. and QIAO, M. (2019). On generalization error bounds of noisy gradient methods for non-convex learning. arXiv preprint arXiv:1902.00621.
- [79] LI, C., SRIVASTAVA, S. and DUNSON, D. B. (2017). Simple, Scalable and Accurate Posterior Interval Estimation. Biometrika.
- [80] LI, Y. and TURNER, R. (2016). Rényi divergence variational inference. *Advances in Neural Information Processing Systems*.
- [81] LIANG, F., MAHONEY, M. and HODGKINSON, L. (2022). Fat—Tailed Variational Inference with Anisotropic Tail Adaptive Flows. *International Conference on Machine Learning*.
- [82] LIN, D. (2013). Online Learning of Nonparametric Mixture Models via Sequential Variational Approximation. Advances in Neural Information Processing Systems.
- [83] LOH, P.-L. and WAINWRIGHT, M. J. (2013). Regularized Mestimators with nonconvexity: Statistical and algorithmic theory for local optima. Advances in Neural Information Processing Systems.
- [84] LOO, N., SWAROOP, S. and TURNER, R. E. (2020). Generalized Variational Continual Learning. *arXiv* preprint *arXiv*:2011.12328.
- [85] LU, X., PERRONE, V., HASENCLEVER, L., TEH, Y. W. and VOLLMER, S. (2017). Relativistic Monte Carlo. *Artificial Intelligence and Statistics*.
- [86] MA, Y.-A., CHEN, T. and FOX, E. (2015a). A complete recipe for stochastic gradient MCMC. Advances in Neural Information Processing Systems.
- [87] MA, Y.-A., CHEN, T. and FOX, E. (2015b). A complete recipe for stochastic gradient MCMC. *Advances in Neural Information Processing Systems*.
- [88] MACLAURIN, D. and ADAMS, R. (2014). Firefly Monte Carlo: exact MCMC with subsets of data. *Conference on Uncertainty in Artificial Intelligence*.
- [89] MAHONEY, M. (2011). Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*.
- [90] MANGOUBI, O., PILLAI, N. S. and SMITH, A. (2018). Does Hamiltonian Monte Carlo mix faster than a random walk on multimodal densities? arXiv preprint arXiv:1808.03230.
- [91] MANOUSAKAS, D., RITTER, H. and KARALETSOS, T. (2022). Black-box coreset variational inference. Advances in Neural Information Processing Systems.

- [92] MANOUSAKAS, D., XU, Z., MASCOLO, C. and CAMPBELL, T. (2020). Bayesian pseudocoresets. Advances in Neural Information Processing Systems.
- [93] MAO, X., LI, Q., XIE, H., LAU, R. Y., WANG, Z. and PAUL SMOLLEY, S. (2017). Least squares generative adversarial networks. Proceedings of the IEEE International Conference on Computer Vision.
- [94] MATHIEU, E. and NICKEL, M. (2020). Riemannian continuous normalizing flows. Advances in Neural Information Processing Systems.
- [95] MEI, S., BAI, Y. and MONTANARI, A. (2018). The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*.
- [96] MESQUITA, D., BLOMSTEDT, P. and KASKI, S. (2020). Embarrassingly parallel MCMC using deep invertible transformations. *Uncertainty in Artificial Intelligence*.
- [97] MINKA, T. and LAFFERTY, J. (2002). Expectation-Propagation for the Generative Aspect Model. *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*.
- [98] MINSKER, S. (2019). Distributed statistical estimation and rates of convergence in normal approximation. *Electronic Journal of Statistics*.
- [99] MINSKER, S., SRIVASTAVA, S., LIN, L. and DUNSON, D. (2014). Scalable and robust Bayesian inference via the median posterior. *International Conference on Machine Learning*.
- [100] MINSKER, S., SRIVASTAVA, S., LIN, L. and DUNSON, D. B. (2017). Robust and scalable Bayes via a median of subset posterior measures. *Journal of Machine Learning Research*.
- [101] MIRZA, M. and OSINDERO, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
- [102] MIYATO, T., KATAOKA, T., KOYAMA, M. and YOSHIDA, Y. (2018). Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.
- [103] MOHASEL AFSHAR, H. and DOMKE, J. (2015). Reflection, refraction, and hamiltonian monte carlo. Advances in Neural Information Processing Systems.
- [104] NAGAPETYAN, T., DUNCAN, A., HASENCLEVER, L., VOLLMER, S., SZPRUCH, L. and ZYGALAKIS, K. (2017). The true cost of stochastic gradient Langevin dynamics. arXiv:1706.02692.
- [105] NAIK, C., ROUSSEAU, J. and CAMPBELL, T. (2022). Fast Bayesian coresets via subsampling and quasi-Newton refinement. *Advances in Neural Information Processing Systems*.
- [106] NEAL, R. M. et al. (2011). MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo.
- [107] NEISWANGER, W., WANG, C. and XING, E. P. (2014). Asymptotically exact, embarrassingly parallel MCMC. Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence.
- [108] NEMETH, C. and FEARNHEAD, P. (2021). Stochastic gradient Markov chain Monte Carlo. *Journal of the American Statistical Association*.
- [109] NEMETH, C. and SHERLOCK, C. (2018). Merging MCMC Subposteriors through Gaussian-Process Approximations. *Bayesian Analysis*.
- [110] NGUYEN, C. V., LI, Y., BUI, T. D. and TURNER, R. E. (2018).
 Variational Continual Learning. International Conference on Learning Representations.
- [111] NI, Y., JI, Y. and MÜLLER, P. (2020). Consensus Monte Carlo for random subsets using shared anchors. *Journal of Computational and Graphical Statistics*.
- [112] NIKOLAKAKIS, K. E., HADDADPOUR, F., KARBASI, A. and KALOGERIAS, D. S. (2022). Beyond Lipschitz: sharp generalization and excess risk bounds for full-batch GD. *arXiv* preprint *arXiv*:2204.12446.

- [113] NING, B. (2021). Spike and slab Bayesian sparse principal component analysis. arXiv preprint arXiv:2102.00305.
- [114] NISHIMURA, A. and DUNSON, D. (2016). Geometrically tempered Hamiltonian Monte Carlo. *arXiv preprint arXiv:1604.00872*.
- [115] NISHIMURA, A., DUNSON, D. and LU, J. (2017). Discontinuous Hamiltonian Monte Carlo for sampling discrete parameters. *arXiv preprint arXiv:1705.08510*.
- [116] Ohn, I. and Lin, L. (2021). Adaptive variational Bayes: Optimality, computation and applications. *arXiv* preprint *arXiv*:2109.03204.
- [117] ONEILL, J. (2020). An overview of neural network compression. arXiv:2006.03669.
- [118] ORBANZ, P. (2017). Subsampling large graphs and invariance in networks. *arXiv:1710.04217*.
- [119] ORBANZ, P. and ROY, D. (2015). Bayesian models of graphs, arrays, and other exchangeable structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [120] OU, R., SEN, D. and DUNSON, D. (2021). Scalable Bayesian inference for time series via divide-and-conquer. arXiv preprint arXiv:2106.11043.
- [121] PAKMAN, A. and PANINSKI, L. (2013). Auxiliary-variable exact Hamiltonian Monte Carlo samplers for binary distributions. Advances in Neural Information Processing Systems.
- [122] PAPAMAKARIOS, G., NALISNICK, E. T., REZENDE, D. J., MOHAMED, S. and LAKSHMINARAYANAN, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*.
- [123] PARIKH, N., BOYD, S. et al. (2014). Proximal algorithms. Foundations and Trends® in Optimization.
- [124] PASARICA, C. and GELMAN, A. (2010). Adaptively scaling the Metropolis algorithm using expected squared jumped distance. *Statistica Sinica*.
- [125] PATI, D., BHATTACHARYA, A. and YANG, Y. (2018). On statistical optimality of variational Bayes. Proceedings of the 21st International Conference on Artificial Intelligence and Statistics.
- [126] PETZKA, H., FISCHER, A. and LUKOVNICOV, D. (2017). On the regularization of Wasserstein GANs. arXiv preprint arXiv:1709.08894.
- [127] PLASSIER, V., VONO, M., DURMUS, A. and MOULINES, E. (2021). DG-LMC: a turn-key and scalable synchronous distributed MCMC algorithm via Langevin Monte Carlo within Gibbs. *International Conference on Machine Learning*.
- [128] PLUMMER, M. (2015). Cuts in Bayesian graphical models. *Statistics and Computing*.
- [129] POMPE, E., HOLMES, C. and LATUSZYŃSKI, K. (2020). A framework for adaptive MCMC targeting multimodal distributions. *The Annals of Statistics*.
- [130] QUIROZ, M., VILLANI, M., KOHN, R., TRAN, M.-N. and DANG, K.-D. (2018). Subsampling MCMC—an introduction for the survey statistician. *Sankhya: The Indian Journal of Statistics*.
- [131] QUIROZ, M., KOHN, R., VILLANI, M. and TRAN, M.-N. (2019). Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*.
- [132] RANGANATH, R., GERRISH, S. and BLEI, D. (2014). Black box variational inference. *International Conference on Artificial Intelligence and Statistics*.
- [133] RASMUSSEN, C. E. and WILLIAMS, C. K. (2006). Gaussian Processes for Machine Learning. *MIT Press*.
- [134] RAY, K., SZABÓ, B. and CLARA, G. (2020). Spike and slab variational Bayes for high dimensional logistic regression. Proceedings of the 34th International Conference on Neural Information Processing Systems.

- [135] RAY, K. and SZABÓ, B. (2022). Variational Bayes for highdimensional linear regression with sparse priors. *Journal of the American Statistical Association*.
- [136] RENDELL, L. J., JOHANSEN, A. M., LEE, A. and WHITE-LEY, N. (2020). Global consensus Monte Carlo. *Journal of Computational and Graphical Statistics*.
- [137] REZENDE, D. and MOHAMED, S. (2015). Variational inference with normalizing flows. *International Conference on Machine Learning*.
- [138] REZENDE, D. J., PAPAMAKARIOS, G., RACANIERE, S., ALBERGO, M., KANWAR, G., SHANAHAN, P. and CRANMER, K. (2020). Normalizing flows on tori and spheres. *International Conference on Machine Learning*.
- [139] ROBERTS, G. O. and ROSENTHAL, J. S. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*.
- [140] ROTH, K., LUCCHI, A., NOWOZIN, S. and HOFMANN, T. (2017). Stabilizing training of generative adversarial networks through regularization. Advances in Neural Information Processing Systems.
- [141] SALIMANS, T., KINGMA, D. and WELLING, M. (2015). Markov chain Monte Carlo and variational inference: bridging the gap. *International Conference on Machine Learning*.
- [142] SAVITSKY, T. D. and SRIVASTAVA, S. (2018). Scalable Bayes under informative sampling. *Scandinavian Journal of Statistics*.
- [143] SCOTT, S., BLOCKER, A., BONASSI, F., CHIPMAN, H., GEORGE, E. and MCCULLOCH, R. (2016a). Bayes and big data: the consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management.*
- [144] SCOTT, S. L., BLOCKER, A. W., BONASSI, F. V., CHIP-MAN, H. A., GEORGE, E. I. and MCCULLOCH, R. E. (2016b). Bayes and big data: the consensus Monte Carlo algorithm. International Journal of Management Science and Engineering Management.
- [145] SHUN, Z. and McCullagh, P. (1995). Laplace approximation of high dimensional integrals. *Journal of the Royal Statistical Society: Series B (Methodological)*.
- [146] SHYAMALKUMAR, N. D. and SRIVASTAVA, S. (2022). An algorithm for distributed Bayesian inference. *Stat.*
- [147] SONG, J., ZHAO, S. and ERMON, S. (2017). A-NICE-MC: Adversarial training for MCMC. *Advances in Neural Information Processing Systems*.
- [148] SRIVASTAVA, S., LI, C. and DUNSON, D. B. (2018). Scalable Bayes via barycenter in Wasserstein space. *Journal of Machine Learning Research*.
- [149] SRIVASTAVA, S., CEVHER, V., DINH, Q. and DUNSON, D. (2015). WASP: Scalable Bayes via barycenters of subset posteriors. Artificial Intelligence and Statistics.
- [150] R CORE TEAM (2021). R: A Language and Environment for Statistical Computing R Foundation for Statistical Computing, Vienna, Austria.
- [151] TRAN, D., VAFA, K., AGRAWAL, K., DINH, L. and POOLE, B. (2019). Discrete flows: Invertible generative models of discrete data. *Advances in Neural Information Processing Systems*.
- [152] UEHARA, M., SATO, I., SUZUKI, M., NAKAYAMA, K. and MATSUO, Y. (2016). Generative adversarial nets from a density ratio estimation perspective. arXiv preprint arXiv:1610.02920.
- [153] VIHOLA, M. (2012). Robust adaptive Metropolis algorithm with coerced acceptance rate. *Statistics and Computing*.
- [154] VONO, M., DOBIGEON, N. and CHAINAIS, P. (2019). Split-and-augmented Gibbs sampler—Application to large-scale inference problems. *IEEE Transactions on Signal Processing*.
- [155] VONO, M., DOBIGEON, N. and CHAINAIS, P. (2020). Asymptotically exact data augmentation: Models, properties, and algorithms. *Journal of Computational and Graphical Statistics*.

- [156] VONO, M., PAULIN, D. and DOUCET, A. (2022). Efficient MCMC sampling with dimension-free convergence rate using ADMM-type splitting. The Journal of Machine Learning Research.
- [157] VYNER, C., NEMETH, C. and SHERLOCK, C. (2023). SwISS: A scalable Markov chain Monte Carlo divide-and-conquer strategy. Stat.
- [158] WAINWRIGHT, M. and JORDAN, M. (2008). Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning.
- [159] WANG, Y., AUDIBERT, J.-Y. and MUNOS, R. (2008). Algorithms for infinitely many-armed bandits. *Advances in Neural Information Processing Systems*.
- [160] WANG, Y. and BLEI, D. (2018). Frequentist consistency of variational Bayes. *Journal of the American Statistical Association*.
- [161] WANG, X. and DUNSON, D. B. (2013). Parallelizing MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605*.
- [162] WANG, C. and SRIVASTAVA, S. (2023). Divide-and-Conquer Bayesian inference in hidden Markov models. *Electronic Journal* of Statistics.
- [163] WANG, W., SUN, Y. and HALGAMUGE, S. (2018). Improving MMD-GAN training with repulsive loss function. arXiv preprint arXiv:1812.09916.
- [164] WANG, X., GUO, F., HELLER, K. A. and DUNSON, D. B. (2015). Parallelizing MCMC with random partition trees. Advances in Neural Information Processing Systems.
- [165] WANG, T., ZHU, J.-Y., TORRALBA, A. and EFROS, A. (2018). Dataset distillation. arXiv:1811.10959.
- [166] WELLING, M. and TEH, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. *International Confer*ence on Machine Learning.
- [167] WILLIAMS, C. and RASMUSSEN, C. (1995). Gaussian processes for regression. Advances in Neural Information Processing Systems.
- [168] WILLIAMS, C. and SEEGER, M. (2001). Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*.
- [169] WINN, J. and BISHOP, C. M. (2005). Variational Message Passing. *Journal of Machine Learning Research*.
- [170] WU, C. and ROBERT, C. P. (2017). Average of Recentered Parallel MCMC for Big Data. arXiv preprint arXiv:1706.04780.
- [171] XUE, J. and LIANG, F. (2019). Double-Parallel Monte Carlo for Bayesian analysis of big data. *Statistics and Computing*.
- [172] YANG, Y. and MARTIN, R. (2020). Variational approximations of empirical Bayes posteriors in high-dimensional linear models. *arXiv preprint arXiv:2007.15930*.
- [173] YANG, Y., PATI, D. and BHATTACHARYA, A. (2020). α-variational inference with statistical guarantees. The Annals of Statistics.
- [174] ZHANG, F. and GAO, C. (2020). Convergence rates of variational posterior distributions. *The Annals of Statistics*.
- [175] ZHANG, J., KHANNA, R., KYRILLIDIS, A. and KOYEJO, O. (2021). Bayesian coresets: revisiting the nonconvex optimization perspective. *Artificial Intelligence in Statistics*.
- [176] ZHOU, J., KHARE, K. and SRIVASTAVA, S. (2022). Asynchronous and Distributed Data Augmentation for Massive Data Settings. *Journal of Computational and Graphical Statistics*.
- [177] ZIEGLER, Z. and RUSH, A. (2019). Latent normalizing flows for discrete sequences. *International Conference on Machine Learning*.