Bayesian Analysis (2022) **TBA**, Number TBA, pp. 1–32

Distributed Computation for Marginal Likelihood based Model Choice[*]

Alexander Buchholz[†,¶], Daniel Ahfock[§,¶], and Sylvia Richardson[‡]

Abstract. We propose a general method for distributed Bayesian model choice, using the marginal likelihood, where a data set is split in non-overlapping subsets. These subsets are only accessed locally by individual workers and no data is shared between the workers. We approximate the model evidence for the full data set through Monte Carlo sampling from the posterior on every subset generating a model evidence per subset. The results are combined using a novel approach which corrects for the splitting using summary statistics of the generated samples. Our divide-and-conquer approach enables Bayesian model choice in the large data setting, exploiting all available information but limiting communication between workers. We derive theoretical error bounds that quantify the resulting trade-off between computational gain and loss in precision. The embarrassingly parallel nature yields important speed-ups when used on massive data sets as illustrated by our real world experiments. In addition, we show how the suggested approach can be extended to model choice within a reversible jump setting that explores multiple feature combinations within one run.

MSC2020 subject classifications: Primary 62F15; secondary 68W15.

Keywords: marginal likelihood, distributed computation.

1 Introduction

The marginal likelihood, also known as the posterior normalising constant or the Bayesian model evidence, is a quantity that is notoriously difficult to calculate, but crucial for model selection in a Bayesian setting (Kass and Raftery, 1995; Robert, 2007). Evaluation of the marginal likelihood is particularly computationally challenging with massive data sets that are too large to fit in the memory of a single computational node. Distributed computing is attractive in such a situation, as the data set can be split into smaller manageable subsets which can be allocated to different nodes and then processed in parallel. Such divide-and-conquer approaches are particularly useful in settings where communication among different workers is costly or limited. Moreover, privacy concerns, governance issues, data security or institutional constraints often make sharing data difficult, a situation which is common when processing medical data. We suggest a novel method for the distributed calculation of the marginal likelihood with large data sets under communication constraints.

[*] This work was supported by the EPSRC (EP/R018561/1), an MRC programme grant (MC_UU_00002/10) and the Alan Turing Institute (TU/B/00092).

[†] Amazon Berlin, Germany, work done while at the University of Cambridge, ab2603@cantab.ac.uk

[‡] MRC Biostatistics Unit, University of Cambridge, UK

[§] School of Mathematics and Physics, The University of Queensland, AUS

[¶] Equal contributions.

https://doi.org/10.1214/22-BA1321

Distributed Bayesian computation largely falls in the MapReduce or split-apply-combine frameworks for computation (Dean and Ghemawat, 2008; Wickham, 2011). The distributed Bayesian computational workflow can be described in three key steps:

- Split: Divide the data set into subsets and distribute across nodes (workers).
- Apply: Each worker independently computes a posterior distribution based on a subset.
- Combine: Aggregate the worker posterior distributions to form a consensus.

The combine step involves the synthesis of multiple probability distributions, and falls under the general umbrella of belief aggregation, a well studied topic in meta-analysis and probabilistic forecasting (West, 1984; Genest, 1984; Dietrich, 2010). Our goal is to determine how marginal likelihood based Bayesian model choice can be operationalised within the aforementioned computational framework. We find that belief aggregation rules for model choice differ in some key aspects compared to fixed model inference and examine the theoretical and computational consequences.

In summary, our main contributions are the following.

- We derive a general decomposition of the marginal likelihood that enables efficient divide-and-conquer calculation without accessing the data in one single place. The combination of the results requires minimal communication and no exchange of data. The computational complexity per worker is $\mathcal{O}(n/S)$ instead of $\mathcal{O}(n)$ on a single machine, where n is the number of observations and S the number of workers.
- We provide a theoretical analysis of two different algorithms for distributed calculation of the marginal likelihood. The first is a simulation consistent approach making use of data augmentation, and the second is an approximate approach relying on local normal approximations. Error bounds are developed for the approximate approach.
- We illustrate the performance of our method on several challenging applications with millions of data points and show that the computation time is reduced by several orders of magnitude, incurring only a negligible bias.
- We show how to apply our approach in a reversible jump setting where a Markov chain Monte Carlo (MCMC) sampler moves between different feature spaces.

The rest of our work is structured as follows. We discuss related work in Section 2 and introduce relevant background material in Section 3. Then we present our main contributions on distributed Bayesian model choice in Section 4. In Section 5 we demonstrate the applicability of our approach on several data sets and models before discussing possible extensions in Section 6.

2 Related Work

Previous work on computationally efficient methods for processing data sets with massive sample sizes can generally be divided into two broad categories. Bardenet et al. (2017), Jahan et al. (2020) and Zhu et al. (2017) provide reviews focusing on Bayesian methods for big data.

A first stream of work focuses on speeding up computation using mini batches (i.e., random subsets) of the entire data. Initially introduced via optimisation in the field of machine learning, the idea of approximating a posterior using mini batches has received substantial attention since the work of Welling and Teh (2011) on stochastic gradient Langevin sampling and the work of Hoffman et al. (2013) on mini batch sampling for variational inference. Subsequent work on this family of algorithms has included theoretical analysis and practical extensions, see, for example, Chen et al. (2014); Alquier et al. (2016); Quiroz et al. (2019); Dang et al. (2019). The calculation of normalising constants, however, has received less attention despite the work of Lyne et al. (2015); Gunawan et al. (2020).

The second line of work aims to make use of parallel processing for reducing computation time. The idea of using a divide-and-conquer approach has seen interest in both the statistics and the machine learning community, see, for example, Deisenroth and Ng (2015) for an application in Gaussian processes and Jordan et al. (2019) for a general approach under communication constraints. A variety of follow-up has been sparked by the work of Scott et al. (2016). The consensus Monte Carlo (CMC) approach performs posterior sampling on data shards and combines the results on a single worker. This approach has been discussed and improved both from a theoretical and practical perspective (Wang and Dunson, 2013; Neiswanger et al., 2013; Minsker et al., 2014; Scott et al., 2017; Srivastava et al., 2018; Zhang et al., 2018; Szabó and van Zanten, 2019). The idea of distributed computation has since been picked up in different communities as, for example, in sequential Monte Carlo (Rendell et al., 2020) and expectation propagation (Gelman et al., 2017; Barthelmé et al., 2018). Combining different models using different data sources has also received substantial attention, see for instance Goudie et al. (2019); Jacob et al. (2017). The concept of federated learning (Li et al., 2020), where the estimation of a model is achieved in a highly distributed setting with repeated rounds of communication between a central node and workers, is another adjacent field to our work. Despite growing interest in Bayesian federated learning (Chen and Chao, 2021; Yurochkin et al., 2019), Bayesian model choice has seen little investigation.

We will proceed under the assumption that there are barriers to implementing a mini-batch based algorithm. Storing the data in one location may be infeasible, or privacy restrictions and communication costs rule out the possibility for one central node to be connected to each site with a subset of the data. Situations where this may occur include the analysis of electronic health records and the analysis of genetic data from large cohorts. It is natural to consider the Laplace approximation to the marginal likelihood given the large sample size (Kass and Raftery, 1995). However, the assumed barriers impede the ability to compute directly the Laplace approximation as determining the posterior mode and Hessian would require extensive communication between sites and the central node, see, for example, McMahan et al. (2017); Safaryan et al. (2021). Divide-and-conquer approaches will still be feasible in such a setting.

3 Background

We first give an overview on Bayesian model choice using the marginal likelihood, and useful computational techniques for estimation of the marginal likelihood. We then discuss core principles for distributed Bayesian inference, and cover the use of normal approximations in the consensus Monte Carlo algorithm by Scott et al. (2016).

3.1 Bayesian model choice

The normalising constant We define the posterior distribution of a parameter $\theta \in \Theta \subset \mathbb{R}^p$ given data y as $p(\theta|y) = p(y|\theta)p(\theta)/p(y)$. It depends on the unknown normalising constant $p(y)$. This normalising constant is the marginal likelihood of the data given the model, also called model evidence, and is calculated as

$$p(y) = \int_\Theta p(y|\theta)p(\theta)d\theta.$$

In most settings this constant is not analytically tractable, as it involves the integration over a potentially high dimensional parameter space Θ. Various sampling based methods are available for the approximate calculation of the evidence, for example, importance sampling (IS) (Geweke, 1989), sequential Monte Carlo (Del Moral et al., 2006), nested sampling (Skilling, 2006) or bridge sampling (Meng and Wong, 1996; Gelman and Meng, 1998). See Knuth et al. (2015) for a review of the different methods.

The Bayes factor The posterior distribution over a set of competing models is defined using the marginal likelihood and a prior distribution on models. Models may be distinguished by different choices of link functions, hyper parameters or feature spaces. The posterior probability of model m_i is given by

$$p(m_i|y) = \frac{p(y|m_i)p(m_i)}{\sum_{k=1}^{K} p(y|m_k)p(m_k)}, \tag{1}$$

where $p(y|m_i) = \int_\Theta p(y|\theta, m_i)p(\theta|m_i)d\theta$, with both prior and likelihood now being dependent on the model i. Models may also be compared using the Bayes factor (BF) (Kass and Raftery, 1995). The BF is calculated as

$$B_{m_1,m_2} = \frac{p(y|m_1)}{p(y|m_2)} = \frac{p(m_1|y)}{p(m_2|y)} \times \frac{p(m_2)}{p(m_1)},$$

where $p(m_i)$ denotes the prior probability of model i, $p(m_i|y)$ denotes the posterior probability of the model given the data and $p(y|m_i)$ for $i = 1, 2$ denotes the probability of the data given the model. For two models the BF allows to select the model with highest posterior probability while adjusting for the prior odds. It is an alternative to standard statistical testing when it comes to model choice.

Data augmentation and conditionally conjugate models Data augmentation has proven to be an important technique for Bayesian computation, facilitating both posterior sampling and marginal likelihood estimation (van Dyk and Meng, 2001; Tanner and Wong, 2010). The rationale is to introduce a latent variable $z \in \mathcal{Z}$ such that the complete data model $p(y, z|\theta)$ is more mathematically and computationally tractable than the marginal model $p(y|\theta) = \int_{\mathcal{Z}} p(y, z, |\theta)\ dz$. Example applications include binary regression models, mixture models and factor analysers (van Dyk and Meng, 2001; Holmes and Held, 2006; Tanner and Wong, 2010). A particularly useful strategy is to construct a sampler on the extended space of $\theta, z \sim p(\theta, z|y)$ and to marginalise out z once a cloud of samples has been generated. This can be an efficient strategy if the full conditionals $p(\theta|y, z)$ and $p(z|y, \theta)$ have a known distribution and allows the construction of a Gibbs sampler. Conditionally conjugate models are obtained when selecting the prior $p(\theta)$ to be conjugate to the complete data likelihood $p(y, z, |\theta)$ (Gelman et al., 2013). Prior $p(\theta)$ and the conditional posterior $p(\theta|y, z)$ are part of the same distribution family and hence sampling from the conditional posterior becomes straightforward for commonly chosen priors. The posterior distribution can be represented as $p(\theta|y) = \int p(\theta|z, y)p(z|y)\ dz$. For any ordinate θ, the log marginal likelihood satisfies

$$\log p(y) = \log p(\theta) + \log p(y|\theta) - \log \int p(\theta|z, y)p(z|y)\ dz$$

Following Chib (Chib, 1995), a simulation consistent estimator of the model evidence is

$$\log \widehat{p}(y) = \log p(\theta) + \log p(y|\theta) - \log \sum_{i=1}^{N} p(\theta|z^i, y)/N,$$

where the z^i are samples from the posterior distribution $p(z|y)$.

3.2 Distributed Bayesian inference

We assume that we observe data $y \in \mathcal{Y}$ consisting of n data points that can be split into S non overlapping data shards y_s, potentially containing several observations such that $y = \{y_1, \cdots, y_S\}$. The likelihood is assumed to satisfy the independence condition $p(y|\theta) = \prod_{s=1}^{S} p(y_s|\theta)$. As outlined in the introduction, the general strategy for distributed Bayesian inference is to allocate a shard of data to each worker who will then compute a local posterior distribution on the basis of the subset y_s. The major challenge in this approach is the formation of a consensus probability distribution given the individual posterior distributions computed by each worker.

An important tool for the synthesis of probability distributions is product-of-experts pooling (Genest, 1984; Hinton, 2002; Dietrich, 2010). Taking θ to be the quantity of interest, the products-of-experts consensus distribution $q(\theta)$ is formed by taking the product of the worker distributions $\{q_s(\theta)\}_{s=1}^{S}$, so $q(\theta) \propto \prod_{s=1}^{S} q_s(\theta)$. The significance of product-of-experts pooling is readily seen when the model is treated as being fixed and known. The full posterior can be represented as

$$p(\theta|y) \propto \prod_{s=1}^{S} p(y_s|\theta)p(\theta)^{1/S}, \tag{2}$$

where $p(y_s|\theta)$ is a likelihood factor over the shard y_s and $p(\theta)^{1/S}$ is the unnormalised subprior, i.e. a fraction of the initial prior $p(\theta)$. Each worker may compute a so-called subposterior $\widetilde{p}(\theta|y_s)$ on a shard of data using the fractionated prior $\tilde{p}(\theta|y_s) \propto p(y_s|\theta)p(\theta)^{1/S}$. The full posterior distribution can then be obtained by product-of-experts pooling of the subposteriors (Huang and Gelman, 2005; Scott et al., 2016)

$$p(\theta|y) \propto \prod_{s=1}^{S} \tilde{p}(\theta|y_s). \tag{3}$$

This important property does not transfer to model selection. There is no analogous product-of-experts decomposition of the full posterior distribution over models

$$p(m_i|y) \not\propto \prod_{s=1}^{S} \tilde{p}(m_i|y_s), \tag{4}$$

where the subposterior model probabilities are defined by $\widetilde{p}(m_i|y_s) \propto \widetilde{p}(y|m_i)p(m_i)^{1/S}$ and the subposterior evidence $\widetilde{p}(y_s|m_i) = \int_\Theta p(y_s|\theta, m_i)\tilde{p}(\theta|m_i)\, d\theta$ is determined using the normalised subprior $\widetilde{p}(\theta|m_i) = p(\theta|m_i)^{1/S} / \int_\Theta p(\theta|m_i)^{1/S} d\theta$. This is due to the fact that in general,

$$p(y|m_i) = \int_\Theta \prod_{s=1}^{S} p(y_s|\theta, m_i)p(\theta|m_i)^{1/S} d\theta \neq \prod_{s=1}^{S} \int_\Theta p(y_s|\theta, m_i)p(\theta|m_i)^{1/S} d\theta.$$

The lack of a product-of-experts decomposition for the full posterior on models (4) suggests that the protocol for distributed Bayesian inference in the fixed model setting may not be effective for model choice. If the goal is to recover the full posterior model probabilities $p(m_i|y)$, it is no longer sufficient to compute subposterior distributions over models in the apply step and then form a consensus distribution using product-of-experts pooling in the combine step. The correct belief aggregation procedure for model selection is developed in Section 4.

In practice, each worker will typically return a Monte Carlo or analytic approximation of the subposterior distribution. An important consideration with distributed Bayesian inference is how the mean squared error of the consensus scales with the number of subsets S (Bardenet et al., 2017; Scott et al., 2016; Neiswanger et al., 2013). There is almost always a trade-off between the error and the computational benefits afforded by distributing the workload across more nodes S. This dynamic will be a key feature in our theoretical analysis and computational experiments.

3.3 Consensus Monte Carlo

The Consensus Monte Carlo algorithm (Scott et al., 2016) is based on the product-of-experts posterior decomposition (3). The key to the approach is that the product of normal subposteriors is proportional to another normal distribution:

$$\prod_{s=1}^{S} \mathcal{N}(\theta|\mu_s, \Sigma_s) \propto \mathcal{N}(\theta|\mu, \Sigma),$$

where the overall variance and mean are obtained as $\Sigma^{-1} = \sum_{s=1}^{S} \Sigma_s^{-1}$ and $\mu = \Sigma \sum_{s=1}^{S} \Sigma_s^{-1} \mu_s$.

The consensus Monte Carlo algorithm samples N points $\theta_s^1, \cdots, \theta_s^N$ from each of the individual subposteriors $\tilde{p}(\theta|y_s)$. This sampling can be achieved, for example, using standard MCMC algorithms like random walk Metropolis-Hasting (Hastings, 1970; Dunson and Johndrow, 2019) or Hamiltonian Monte Carlo (Neal et al., 2011). The samples are then recombined using a normal approximation to the subposteriors.

The normal approximation $\mathcal{N}(\mu_s, \Sigma_s) \approx \tilde{p}(\theta|y_s)$ is based on the estimated mean μ_s and variance Σ_s from the subposterior samples, using the Laplace-Metropolis approximation (Lewis and Raftery, 1997). This approximation is asymptotically justified through the Bernstein-von-Mises (BvM) theorem (see also Ghosh and Ramamoorthi (2003)). The recombination of the samples from the local Markov chains is based on a weighting according to the inverse covariances of the subposteriors (Scott et al., 2016). Combining the sampling based approach with the normal approximation has the advantage that more features of the posterior distribution are captured compared to the use of a plain normal approximation where the sampling step would be omitted. See Scott et al. (2016, 2017) for more details.

4 Distributed Model Choice

We will now introduce and discuss our contributions making use of the previously introduced background material.

Decomposing the model evidence Using Bayes' Theorem and some elementary algebra, it is possible to obtain an identity for the marginal likelihood that lends itself to distributed computation.

Proposition 1. *The model evidence for the full data can be decomposed as*

$$p(y) = \alpha^S \prod_{s=1}^{S} \tilde{p}(y_s) \int_\Theta \prod_{s=1}^{S} \tilde{p}(\theta|y_s) \; d\theta, \tag{5}$$

where $\tilde{p}(\theta) = \frac{p(\theta)^{1/S}}{\alpha}$ is the normalised subprior, $\alpha_\Theta = \int p(\theta)^{1/S} d\theta$ is the normalising constant of the subprior, $\tilde{p}(y_s) = \int_\Theta p(y_s|\theta)\tilde{p}(\theta)d\theta$ is the normalising constant of the subposterior and $\tilde{p}(\theta|y_s) = p(y_s|\theta)\tilde{p}(\theta)/\tilde{p}(y_s)$ denotes the normalised subposterior.

Proof. See Appendix (Buchholz et al., 2022). □

Each of the three components in the decomposition (5) has an interpretation in terms of the generic split-apply-combine framework mentioned in the introduction. The term α^S reflects that each worker is allocated a fraction of the prior information in the split step. The subposterior evidence for each shard $\tilde{p}(y_s)$ is computed by the workers in the apply stage. The evaluation of the subposterior integral $\int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s)d\theta$ is a necessary step for appropriate evidence synthesis in the combine stage.

Using Proposition 1 it is straightforward to show that the full posterior distribution on models has the modified product-of-experts representation

$$p(m_i|y) \propto \left\{ \prod_{s=1}^S \tilde{p}(m_i|y_s) \right\} \alpha_i^S \left(\int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s, m_i)\ d\theta \right), \tag{6}$$

recalling that the subposterior probabilities are given by $\widetilde{p}(m_i|y_s) \propto \widetilde{p}(y_s|m_i)p(m_i)^{1/S}$. Comparing (6) to (4), we see that a modified product-of-experts rule is necessary for distributed Bayesian model selection, as the subposterior model probabilities $\tilde{p}(m_i|y_s)$ do not contain enough information to recover the full posterior model probabilities $p(m_i|y)$. Subset analyses will be under-powered relatively compared to the full data set analysis if the shard size is small compared to the total sample size, and this may manifest in a bias towards smaller models in the model subposteriors $\tilde{p}(m_i|y_s)$. A secondary issue is that a model may appear to fit well in each subset but be of poor fit overall. The inclusion of the integral term over the subposteriors $\int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s, m_i)\ d\theta$ allows the global goodness of fit to be reconstructed by considering the overlap in the subposterior distributions.

We will now focus on how to use the presented decomposition in an algorithm to approximate the marginal likelihood of the full data set. In a variety of settings α can be computed analytically. For example, if $p(\theta)$ is a normal distribution, fragmenting the prior amounts to an inflation of the prior variance: $\mathcal{N}(0,\Sigma)^{1/S} \propto \mathcal{N}(0, S\Sigma)$ and α is thus obtained easily. The same holds true for a Laplace prior, where $\mathcal{L}(0,\sigma)^{1/S} \propto \mathcal{L}(0, S\sigma)$. However, care must be taken for some distributions of the exponential family. When S is too large, the integral $\int_\Theta p(\theta)^{1/S}d\theta$ becomes infinite as too much mass is pushed into the tails. We recommend checking this on a case by case basis. The local evidence $\tilde{p}(y_s)$ can be calculated by every individual worker using one of the various techniques described at the start of Section 3.1. The most difficult part of (5) is to evaluate the last integral

$$I_{\text{sub}} := \int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s)\ d\theta. \tag{7}$$

We will now discuss two different strategies to estimate the integral I_{sub} (7), an exact approach based on data augmentation and Gibbs sampling, and an approximate approach using normal approximations.

4.1 Data augmentation to compute I_{sub}

A data augmentation strategy can be used to construct a simulation consistent estimator of the components in (5). See Ahfock (2019, Chap 2) for a detailed discussion. In particular, the latent variables are helpful to calculate an estimator of (7) as we show now. We assume that the latent variables z_s are independent given θ. Then, the augmented full data set posterior is again proportional to the subposteriors, i.e.,

$$\begin{aligned} p(\theta|y_{1:S}, z_{1:S}) &\propto p(\theta)p(y_{1:S}, z_{1:S}|\theta) \propto \prod_{s=1}^{S} p(y_s, z_s|\theta)p(\theta)^{1/S}, \\ &\propto \prod_{s=1}^{S} p(y_s, z_s|\theta)\tilde{p}(\theta) \propto \prod_{s=1}^{S} \tilde{p}(\theta|y_s, z_s). \end{aligned}$$

The conditional latent variable subposteriors on the individual workers (accessing only y_s) are given as as

$$\tilde{p}(z_s|y_s,\theta) \propto p(y_s, z_s|\theta)\tilde{p}(\theta) \propto p(z_s|y_s,\theta)p(y_s|\theta) \propto p(z_s|y_s,\theta),$$

and hence the subprior $\tilde{p}(\theta)$ is not material when conditioning on θ. This establishes a Gibbs sampling strategy for sampling from $\tilde{p}(\theta, z_s|y_s)$ using the conditionals defined above. In conditionally conjugate models, where the subpriors are chosen such that $\int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s, z_s)d\theta$ has a closed form solution, we can exploit the samples generated from the latent variable subposteriors to approximate I_{sub}.

Proposition 2. *Using a data augmentation scheme for the augmented data likelihood* $p(y_{1:S}, z_{1:S}|\theta)$ *we have*

$$I_{\text{sub}} = \mathbf{E}_{\tilde{p}(z_{1:S}|y_{1:S})}\left[\int_\Theta \prod_{s=1}^{S} \tilde{p}(\theta|y_s, z_s)d\theta\right], \tag{8}$$

where $\tilde{p}(z_{1:S}|y_{1:S}) = \prod_{s=1}^S \tilde{p}(z_s|y_s)$.

Proof. See Appendix (Buchholz et al., 2022). □

As a a consequence, Proposition 2 suggests the following Monte Carlo estimator

$$\hat{I}_{\text{sub}} = \frac{1}{N}\sum_{i=1}^{N}\int_\Theta \prod_{s=1}^{S} \tilde{p}(\theta|y_s, z_s^i)d\theta, \tag{9}$$

where $\int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s, z_s^i)d\theta$ is calculated analytically and z_s^i are samples obtained from, e.g., a Gibbs sampler. See the algorithmic version of the sampler in Algorithm 1. We provide further details on data augmentation for logistic regressions using Pólya-Gamma data augmentation in the appendix. The variance of the proposed estimator can be understood through a connection to importance sampling.

Proposition 3. *Assume i.i.d. sampling from the latent variable posterior. The variance of the estimator suggested in* (9) *is then*

$$\operatorname{Var}\hat{I}_{\text{sub}} = \frac{I_{\text{sub}}^2}{N}\operatorname{Var}_{\tilde{p}(z_{1:S}|y_{1:S})}\left[\frac{p(z_{1:S}|y_{1:S})}{\prod_{s=1}^S \tilde{p}(z_s|y_s)}\right].$$

Proof. See Appendix (Buchholz et al., 2022). □

This links the variance of the estimator $\hat{I}_{\text{sub}}$ to the relative coverage of the joint conditional posterior $p(z_{1:S}|y_{1:S})$ and the product of the conditional subposteriors $\tilde{p}(z_{1:S}|y_{1:S})$. The variance of the ratio measures the quality of the product of the conditional subposterior as an importance sampling proposal distribution. If the tails of the proposal are thinner than the target, the variance of the estimator potentially becomes infinite. As the number of splits increases, we expect the product of the conditional subposteriors to become a worse approximation of the true subposterior, and the variance of the estimator to increase. Note that in practice the variance of $\hat{I}_{\text{sub}}$ will be larger than the suggested quantity due to the autocorrelation introduced by the Gibbs sampler.

Algorithm 1 Distributed model evidence computation using data augmentation.

```
Input: data $y$, number of chunks $S$, likelihood $p(\cdot|\theta)$, prior $p(\theta)$.
Split: Divide data in $S$ chunks $y_1, \cdots, y_S$.
Apply in parallel:
for $s = 1$ to $S$ do
  Sample $(\theta_s^1, z_s^1), \cdots, (\theta_s^N, z_s^N) \sim \tilde{p}(\theta, z|y_s)$ using Gibbs sampling.
  Calculate and store $\tilde{p}(y_s) = \int p(y_s|\theta)\tilde{p}(\theta)d\theta$ using Chib's method.
end for
Combine:
Calculate $p(y) = \alpha^S \prod_{s=1}^S \tilde{p}(y_s) \int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s)d\theta$,
using $\int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s)d\theta = \mathbf{E}_{\tilde{p}(z_{1:S}|y_{1:S})}\left(\int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s, z_s)d\theta\right) \approx (9)$.
```

4.2 A normal approximation to I_{sub}

Models relying on data augmentation are a rather restricted class of models that require tailor-made samplers. A more widely applicable approach to estimate I_{sub} (7) is a normal approximation to each subposterior $\widetilde{p}(\theta|y_s) \approx \mathcal{N}(\theta|\mu_s, \Sigma_s)$, yielding the following approximation of the integral:

$$I_{\text{sub}} = \int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s)\ d\theta \approx \int_\Theta \prod_{s=1}^S \mathcal{N}(\theta \mid \mu_s, \Sigma_s)\ d\theta.$$

The mean and covariance parameters for the subposterior normal approximations, μ_s, Σ_s can be estimated given subposterior samples generated by each worker. Using the fact that the product of normal density functions is proportional to another normal density we can provide the closed form expression

$$\int_\Theta \prod_{s=1}^S \mathcal{N}(\theta|\mu_s, \Sigma_s)\ d\theta = \exp\left(\sum_{s=1}^S \xi_s - \xi\right), \tag{10}$$

where

$$\eta_s = \Sigma_s^{-1}\mu_s, \Lambda_s = \Sigma_s^{-1},$$

Algorithm 2 Distributed model evidence computation using normal approximations.

```
**Input:** data $y$, number of chunks $S$, likelihood $p(\cdot|\theta)$, prior $p(\theta)$.
**Split**: Divide data in $S$ chunks $y_1, \cdots, y_S$.
**Apply** in parallel:
**for** $s = 1$ **to** $S$ **do**
    Sample $\theta_s^1, \cdots, \theta_s^N \sim \tilde{p}(\theta|y_s)$.
    Calculate and store $\tilde{p}(y_s) = \int p(y_s|\theta)\tilde{p}(\theta)d\theta$ and $\mu_s, \Sigma_s$.
**end for**
**Combine:**
Calculate $\alpha^S \prod_{s=1}^S \tilde{p}(y_s) \int \prod_{s=1}^S \mathcal{N}(\theta|\mu_s, \Sigma_s)d\theta \approx p(y)$.
```

$$\xi_s = -\frac{1}{2}(p \log 2\pi - \log|\Lambda_s| + \eta_s^t \Lambda_s^{-1} \eta_s),$$

$$\eta = \sum_{s=1}^S \Sigma_s^{-1}\mu_s, \ \ \Lambda = \sum_{s=1}^S \Sigma_s^{-1},$$

$$\xi = -\frac{1}{2}(p \log 2\pi - \log|\Lambda| + \eta^t \Lambda^{-1} \eta).$$

The normal approximations to the subposterior $\widetilde{p}(\theta|y_s) \approx \mathcal{N}(\theta|\mu_s, \Sigma_s)$ can be justified under the usual conditions for a Bernstein-von-Mises theorem to hold, see e.g. Ghosh and Ramamoorthi (2003). Due to the involved matrix inversions and calculation of determinants the complexity of calculating (10) is $O(Sp^3)$. We approximate the normalising constants $\tilde{p}(y_s)$ using any of the techniques discussed at the beginning of Section 3.1. Note that the normal approximation is only required for estimating I_{sub}, and does not influence the estimation of $\widetilde{p}(y_s)$. We hence suggest Algorithm 2 for the estimation of the marginal likelihood.

Using mixtures of normals If the posterior distribution is multimodal, a simple normal approximation will result in a poor approximation of the integral of the product of the subposteriors. As noted by Neiswanger et al. (2013); Scott et al. (2017), a mixture of normals or kernel density estimators can be used to approximate the posterior. These approaches have been shown to work well in practice. However, if the number of mixture components or the number of splits S gets large this approach may become prohibitive. As we have to calculate the product of S mixtures, the resulting calculation of the product of distributions has a complexity of $\mathcal{O}(S^k)$, where k denotes the number of mixture components. We therefore refrain from this approach.

Error of the approximation Proposition 4 gives an exact representation of the relative error when using the normal approximation to the subposterior integral I_{sub} (10).

Proposition 4. *Suppose $\{\mu_s, \Sigma_s\}_{s=1}^S$ are the selected parameters for the subposterior normal approximations $\tilde{p}(\theta|y_s) \approx \mathcal{N}(\theta|\mu_s, \Sigma_s)$, and that $\mathcal{N}(\theta|\mu, \Sigma) \propto \prod_{s=1}^S \mathcal{N}(\theta|\mu_s, \Sigma_s)$ is the resulting normal approximation to the full posterior $p(\theta|y)$. Then the proposed* *normal approximation to* I_{sub} *satisfies:*

$$\int_{\Theta} \prod_{s=1}^{S} \widetilde{p}(\theta \mid y_s)\ d\theta = \left\{ \int_{\Theta} \prod_{s=1}^{S} \mathcal{N}(\theta \mid \mu_s, \Sigma_s)\ d\theta \right\} \mathbb{E}_{\mathcal{N}(\theta;\mu,\Sigma)} \left[\prod_{s=1}^{S} \frac{\widetilde{p}(\theta \mid y_s)}{\mathcal{N}(\theta|\mu_s, \Sigma_s)} \right].$$

Proof. See Appendix (Buchholz et al., 2022). □

Proposition 5 provides an error bound for the proposed approximation to the marginal likelihood under assumptions on the quality of the subposterior normal approximations.

Proposition 5. *Suppose* $\{\mu_s, \Sigma_s\}_{s=1}^S$ *are the selected parameters for the subposterior normal approximations* $\tilde{p}(\theta|y_s) \approx \mathcal{N}(\theta|\mu_s, \Sigma_s)$*, and that* $\mathcal{N}(\theta|\mu, \Sigma) \propto \prod_{s=1}^S \mathcal{N}(\theta|\mu_s, \Sigma_s)$ *is the resulting normal approximation to the full posterior* $p(\theta|y)$*. The exact marginal likelihood* $p(y)$ *and the proposed approximation to the marginal likelihood* $\widehat{p}(y)$ *are respectively,*

$$p(y) = \alpha^S \left(\prod_{s=1}^S \widetilde{p}(y_s) \right) \int_{\Theta} \prod_{s=1}^S \tilde{p}(\theta \mid y_s)\ d\theta,$$
$$\widehat{p}(y) = \alpha^S \left(\prod_{s=1}^S \widetilde{p}(y_s) \right) \int_{\Theta} \prod_{s=1}^S \mathcal{N}(\theta \mid \mu_s, \Sigma_s)\ d\theta.$$

Suppose that the subposterior normal approximations satisfy the density ratio bounds

$$\max_{s=1,\ldots,S} \sup_{\theta\in\Theta} \frac{\widetilde{p}(\theta \mid y_s)}{\mathcal{N}(\theta; \mu_s, \Sigma_s)} \le A, \qquad \max_{s=1,\ldots,S} \sup_{\theta\in\Theta} \frac{\mathcal{N}(\theta; \mu_s, \Sigma_s)}{\widetilde{p}(\theta \mid y_s)} \le B.$$

Then the relative error of the proposed approximation to the marginal likelihood satisfies

$$-S \log B + \log \mathbb{E}_{\mathcal{N}(\theta;\mu,\Sigma)}[\mathbb{1}(\theta \in \Theta)] \le \log \frac{p(y)}{\widehat{p}(y)} \le S \log A.$$

Proof. See Appendix (Buchholz et al., 2022). □

If the parameter space is unconstrained, so $\Theta = \mathbb{R}^{\dim(\theta)}$, then $\mathbb{E}_{\mathcal{N}(\theta;\mu,\Sigma)}[\mathbb{1}(\theta \in \Theta)] = 1$ and the bound can be simplified to $-S \log B \le \log p(y)/\widehat{p}(y) \le S \log A$. The supremum of the density ratio is an interesting divergence measure for two probability distributions, with important connections to the total variation distance (Dümbgen et al., 2021). Under the conditions of the Bernstein-von-Mises theorem, the quality of each subposterior normal approximation is expected to improve as the shard size increases. With a fixed number of shards S, A and B will tend to 1 as n increases if $\{\mu_s, \Sigma_s\}_{s=1}^S$ are taken to be the true subposterior means and variances. Corollary 1 provides an error bound under an assumption about the average accuracy of the subposterior normal approximations over the subsets S, rather than the worst case accuracy over the subsets S, as in Proposition 5.

Corollary 1. *Fix the shard size n_s, and let the total sample size be given by $n = Sn_s$. Let $p(y)$ represent the true marginal likelihood and $\widehat{p}(y)$ denote the proposed approximation as in Proposition 5. Assume that*

$$\sup_{\theta\in\Theta}\left|\frac{1}{S}\sum_{s=1}^{S}\log\frac{\widetilde{p}(\theta|y_s)}{\mathcal{N}(\theta|\mu_s,\Sigma_s)}\right| = \mathcal{O}_p(1)$$

as the number of shards S increases, where $\{y_s,\mu_s,\Sigma_s\}_{s=1}^S$ are treated as random variables. Then the relative error satisfies

$$\log\frac{p(y)}{\widehat{p}(y)} = \mathcal{O}_p(S).$$

Proof. See Appendix (Buchholz et al., 2022). □

Proposition 5 and Corollary 1 suggest a trade-off between the computational gains from the proposed distributed strategy via the number of splits S and the resulting approximation error.

In practice we transform (5) to the log domain and get

$$\log\widehat{p}(y) = S\log\alpha + \sum_{s=1}^{S}\log\tilde{p}(y_s) + \log\int\prod_{s=1}^{S}\mathcal{N}(\theta\mid\mu_s,\Sigma_s)d\theta.$$

The corresponding estimator is a sum of $S+2$ terms so the variance of this quantity grows linearly in S. Moreover, when estimating $\log\tilde{p}(y_s)$ by N samples from a Markov chain the MSE is typically of order $\mathcal{O}(1/N)$. Thus, more simulations can reduce this error. In summary, our results suggest that if the error that comes from MCMC sampling is relatively small and that the shard sizes are large enough so that the quality of the subposterior normal approximation is reasonable, our suggested approach will result in good approximations of the full data set marginal likelihood.

4.3 Model choice using reversible jump

Calculation of the marginal likelihood for every model in the candidate set is only feasible when there are a small number of models under consideration. In many practical settings, the aim is often to choose among a large number of models, that potentially have different support. An important example is variable selection, where a particular model consists of a specific combination of selected variables.

The reversible jump approach (Green, 1995) allows the construction of a Markov Chain that jointly traverses models and the associated parameter spaces. The posterior probability of a model given the data $p(m_i|y)$, see (1), is obtained by the relative time the sampler spends exploring that model.

Distributed reversible jump Markov chain Monte Carlo (RJMCMC) The modified product-of-experts belief aggregation rule for models (6)

$$p(m_i|y) \propto \left\{ \prod_{s=1}^{S} \tilde{p}(m_i|y_s) \right\} \alpha_i^S \left(\int_\Theta \prod_{s=1}^{S} \tilde{p}(\theta|y_s, m_i)\ d\theta \right)$$

illuminates how one may conduct distributed Bayesian model selection using reversible jump methodology. During the apply stage, it is sufficient for workers to compute subposterior model probabilities $\widetilde{p}(m_i|y_s) \propto \tilde{p}(y_s|m_i)p(m_i)^{1/S}$, and model subposteriors $\tilde{p}(\theta|y_s, m_i) \propto p(y_s|\theta, m_i)p(\theta|m_i)^{1/S}$. The subposterior integral $\int_\Theta \prod_{s=1}^{S} \tilde{p}(\theta|y_s, m_i)\ d\theta$ can be evaluated in the combine stage, and the term α_i^S can be determined in the split stage. The belief aggregation rule (6) can then be used to reconstruct the full posterior model probabilities. Once again, a normal approximation to each subposterior $\tilde{p}(\theta|y_s, m_i) \approx \mathcal{N}(\theta|\mu_s^i, \Sigma_s^i)$ can be used to approximate the integral

$$\int_\Theta \prod_{s=1}^{S} \tilde{p}(\theta|y_s, m_i)\ d\theta \approx \int_\Theta \prod_{s=1}^{S} \mathcal{N}(\theta|\mu_s^i, \Sigma_s^i) d\theta.$$

Reversible jump is particularly well suited to to the decomposition (6) as the sampler simultaneously explores models and the associated parameter spaces. As such, estimates of $\tilde{p}(m_i|y_s), \mu_s^i$ and Σ_s^i will be readily available from the RJMCMC output of each worker. Consequently, the splitting approach can effectively be combined with a reversible jump algorithm as we suggest in Algorithm 3. See also the appendix (Buchholz et al., 2022) for more details.

Algorithm 3 RJMCMC distributed model choice.

```
Input: data $y$, number of chunks $S$, set of models $m_k$ for $k = 1, \cdots, K$, likelihoods
$p(\cdot|\theta, m_k)$, priors $p(\theta|m_k), p(m_k)$.
Split: Divide data in $S$ chunks $y_1, \cdots, y_S$.
Apply in parallel:
for $s = 1$ to $S$ do
    Sample $\theta_s^1, \cdots, \theta_s^N | m_k \sim \tilde{p}(\theta|y_s, m_k)$ using RJMCMC over all $k$.
    Calculate and store $\tilde{p}(m_k|y_s)$, $\mu_s^k, \Sigma_s^k$ for models $k = 1, \ldots, K$.
end for
Combine:
Calculate $\widehat{p}(m_k|y) \propto \left\{ \prod_{s=1}^{S} \tilde{p}(m_k|y_s) \right\} \alpha_k^S \left( \int_\Theta \prod_{s=1}^{S} \mathcal{N}(\theta|\mu_s^k, \Sigma_s^k)\ d\theta \right)$.
Bayes factors can be estimated as $\widehat{p}(m_k|y)p(m_{k'})/\{\widehat{p}(m_{k'}|y)p(m_k)\}$ for any two mod-
els $(k, k')$.
```

There are a number of known issues with RJMCMC that may limit its effectiveness within a divide-and-conquer approach. Reversible jump samplers are known to be hard to tune and often slow to converge. The construction of suitable proposal distributions can also be very challenging when models are not nested. Moreover, the models of interest have to have been visited a sufficient number of times to get reliable estimates.

All these combined make the use of RJMCMC burdensome. With regards to distributed computation, an issue is that if the space of potential models is large, not all models of interest might have been explored on every data shard.

5 Experiments

In our experiment section we investigate the following questions. (a) Are naive voting strategies, upsampled likelihoods and CMC based importance sampling an alternative to our approach? (b) How does the approach based on data augmentation and the approach based on normal approximations perform relatively? (c) What is the magnitude of the error introduced from approximating I_{sub} compared to the overall scale of the log marginal likelihood? (d) How does the error of I_{sub} behave with respect to the number of splits? (e) What are practical gains from using the distributed approach on very large data sets? (f) How reliable is the distributed approach for RJMCMC? We answer these questions by assessing our approach with a Bayesian logistic regression, a normal linear regression and a linear regression with Laplace priors.

Our overall experimental set up is the following: prior variances on the model parameters are set to 1 and their means to 0. We use a randomised splitting procedure. This means for S splits we will have roughly n/S samples per split, where we use a uniform sampling scheme without replacement, if not otherwise stated. We run every sampler 20 times where at every iteration the splitting and the Monte Carlo sampler are initialised with a different seed. Thus, the observed variation in the outcome is a combination of the variation through splitting (i.e. different partitions) and Monte Carlo sampling. The number of generated MCMC samples per chain is $10,000$ where the first $2,000$ samples are discarded as burn-in. We assess the error of the estimation of the log marginal likelihood by comparing it to the result of the estimation on the whole data set, if this is computationally feasible in a reasonable amount of time.

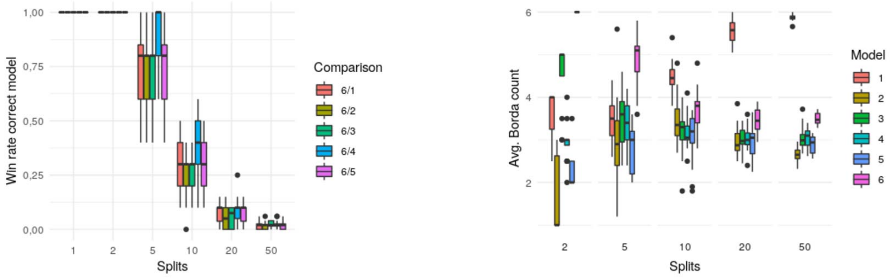
We use the RMSE (root mean squared error), defined as

$$\sqrt{\text{MSE}} = \sqrt{\mathbf{E}\| \log p(y) - \log \hat{p}(y)\|^2},$$

where $\hat{p}(y)$ is approximated using our suggested decomposition. The relative RMSE is defined as $\%\sqrt{\text{MSE}} = \frac{\sqrt{\mathbf{E}\| \log p(y) - \log \hat{p}(y)\|^2}}{\log p(y)} \times 100$. The use of the relative RMSE is justified through the scale of the actual quantity we are trying to estimate.

Practical considerations We implement our algorithm using two generic *R* packages, namely *rstan* (Carpenter et al., 2017) and the *bridgesampling* package (Gronau et al., 2020). *rstan* allows convenient sampling from the posterior distribution of a model using HMC. The package can handle a variety of different models and the user has to provide only a simple code that describes the model. Tuning of hyper parameters and convergence checking is handled automatically. The *bridgesampling* package can use *rstan* models to calculate an approximation of the model evidence. For the reversible jump illustration we use the *R2BGLiMS* package.[1] This package performs model choice

[1] https://github.com/pjnewcombe/R2BGLiMS.



(a) Majority based voting using local marginal likelihoods

(b) Borda counts using local marginal likelihoods

Figure 1: Voting schemes based on local marginal likelihoods. The local marginal likelihoods $\tilde{p}(y_s)$ are computed on all subsets s for 6 different models. In Figure 1a the local marginal likelihood of the correct model 6 is compared with the other local marginal likelihoods. Based on a majority vote it is decided whether model 6 wins. We indicate the win rate of model 6, shown on the y-axis. In Figure 1b we use a Borda voting scheme to aggregate the order of local marginal likelihoods into a global model ordering. We display the average Borda count and hence the correct model 6 must get the highest average count to win. As the number of splits increase (x-axis) the correct model 6 is not chosen anymore neither for the majority voting scheme nor for the Borda count.

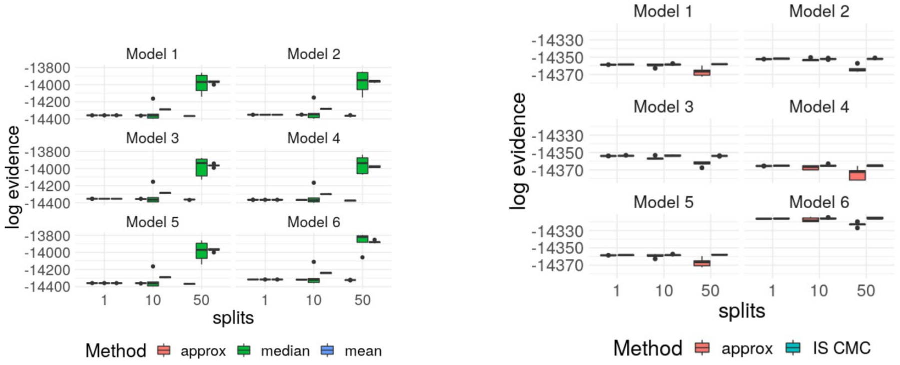
using reversible jump MCMC on logistic, normal and Weibull regression models. We explicitly run the experiment on single core architectures to illustrate the advantages of distributing the computation over a large number of small workers. Code for reproducing the results is available through the first author's github repository.[2]

5.1 Experiment 1: Voting schemes based on local marginal likelihoods and CMC IS

In this section we illustrate potential issues of natural alternatives to our approach based on the decomposition in (5). Although this comparison is limited in scope, we believe it motivates and justifies the in-depth study of Algorithm 2. This experiment is based on six different Gaussian regression models with a log normal prior on the variance that makes this model non-conjugate. We simulate a dataset with 10,000 observations and induce high correlation of the features. The correct model (6) uses the all 17 covariates whereas the other models all omit one relevant variable (see also the appendix (Buchholz et al., 2022) for further details).

Majority voting and Borda counts To motivate the need for a coherent way of combining local evidences, we illustrate what can go wrong when using a naive approach for combining local inference. We try to identify the best model based on the local marginal likelihoods $\tilde{p}(y_s|m_k)$ for the models m_k. We compare two different voting schemes

[2] https://github.com/alexanderbuchholz/distbayesianmc.



(a) Comparison of three different methods for computing the marginal likelihood (y-axis). The boxes inside a number of splits (x-axis) are ordered as left our method (approx), middle the median and right the mean of the upsampled marginal likelihood.

(b) Comparison of the consensus Monte Carlo IS (CMC IS, right most boxes) approach for computing the marginal likelihood (y-axis) with our approximate method (approx, left most boxes) for a varying number of splits (x-axis).

Figure 2: In Figure 2a (left) we compare the upsampled methods {mean, median} with our approximate method (approx) based on Algorithm 2. The mean and median based approaches become unstable with 50 splits of the data. In Figure 2b (right) CMC IS stays competitive as the number of splits increase and seems more stable than our approximate method (approx) based on Algorithm 2. However, an additional round of communication between the central node and the distributed workers is required.

that aggregate local rankings. First, we use a majority voting scheme, where each worker returns the winning model based on local marginal likelihoods, i.e., $\text{winner}_s = \arg\max_{m_k} \tilde{p}(y_s|m_k) \forall s$. Then, the central node aggregates the vote share of model 6 (i.e., on how many workers the correct model won): win rate $= \sum_{s=1}^{S} 1_{\{\text{winner}_s = 6\}}/S$. As a second voting scheme we use Borda counts (Emerson, 2013). The local models receive a score based on their ranking going from 1 to 6, i.e., $\text{Borda score}_{s,m_k} = \arg\text{sort}_{m_k} \tilde{p}(y_s|m_k)$, and the winning model is obtained by averaging the Borda scores $\sum_{s=1}^{S} \text{Borda score}_{s,m_k}/S$. This voting scheme favors a consensus vote over a majority based decision.

As illustrated in Figure 1, the voting schemes work reasonably well for a small number of splits. But with as little as 10 splits (every worker sees 10% of the data), the correct model is not chosen anymore, neither for the majority based voting nor for the Borda counts. Therefore, we decide not to compare our suggested method with these naive aggregations in the remainder of our experiments.

Mean and median of upsampled normalising constants We use the same setting as above and compare two additional methods with our suggested approach from Algorithm 2. We estimate locally the marginal likelihood of what we call an upsampled model following the idea in Zhang et al. (2018), i.e., we replicate locally the data of each shard to match the full data set size. The upsampled marginal likelihood is computed using the replicated data as

$$\acute{p}(y_s) = \int_\Theta p(y_s|\theta)^S p(\theta) d\theta.$$

A final estimator of the marginal likelihood is obtained by computing the mean or median

$$\widehat{p}(y) = \begin{cases} \text{mean}(\{\acute{p}(y_1), \dots, \acute{p}(y_S)\}), \\ \text{med}(\{\acute{p}(y_1), \dots, \acute{p}(y_S)\}), \end{cases}$$

of the local marginal likelihoods when combining them on the central node. We illustrate the results of this approach in Figure 2a. Using medians or means of upsampled local marginal likelihoods becomes unstable when reaching 50 shards compared to our approach in Algorithm 2. We therefore do not consider this method in the rest of our experiments.

Consensus Monte Carlo based importance sampling As a final point of comparison, we suggest a straightforward extension of CMC to the computation of the normalising constant based on importance sampling using the consensus normal distribution. This approach is based on the following importance sampling identity that can also be derived from Proposition 4.

$$p(y) = \int_\Theta \prod_{s=1}^S \frac{p(y_s|\theta)p(\theta)^{1/S}}{\mathcal{N}(\theta|\mu_s, \Sigma_s)} \prod_{s=1}^S \mathcal{N}(\theta|\mu_s, \Sigma_s) d\theta \approx \frac{1}{N} \sum_{i=1}^N \prod_{s=1}^S \underbrace{\frac{p(y_s|\theta_i)p(\theta_i)^{1/S}}{\mathcal{N}(\theta_i|\mu_s, \Sigma_s)}}_{=:w_s(\theta_i)} \times \gamma, \quad (11)$$

where we sample $\theta_i \sim \mathcal{N}(\theta|\mu, \Sigma) \propto \prod_{s=1}^S \mathcal{N}(\theta_i|\mu_s, \Sigma_s)$ for N samples and

$$\gamma = \int_\Theta \prod_{s=1}^S \mathcal{N}(\theta|\mu_s, \Sigma_s) d\theta,$$

correctly normalizes the importance sampling estimator in (11). Thus, we construct an approximation of the whole posterior but we only need to evaluate the ratio of the local subposteriors and the local normal approximation $w_s(\theta_i)$. This approach adds an additional round of communication with the central node, as $\theta_i \sim \mathcal{N}(\theta|\mu, \Sigma)$ is generated on the central node, the weights $w_s(\theta_i)$ are computed locally and then send back to the central node to compute the aggregation. This IS approximation is exact, but can suffer from high variance if the local normal approximation $\mathcal{N}(\theta|\mu_s, \Sigma_s)$ and the consensus approximation $\mathcal{N}(\theta|\mu, \Sigma)$ have little overlap. The product of importance weighting factors $\prod_{s=1}^S w_s(\theta_i)$ potentially introduces further variance as the number of splits S increase (although we did not observe this empirically in Figure 2b).

We illustrate the result of this method in Figure 2b. Using an IS approximation based on CMC with 1000 importance samples performs reasonably well in this setting. A more detailed comparison of the CMC IS approximation for computing the marginal likelihood is out of scope for this work.

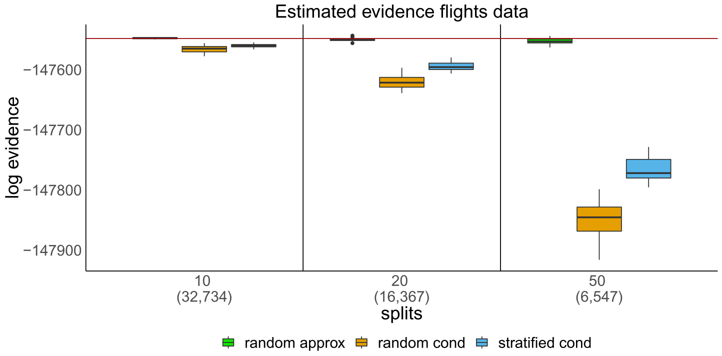


Figure 3: Comparison of the calculated normalising constant (y-axis) for a logistic regression on the flights data. As the number of splits increase (x-axis), the estimates become unreliable for the conditionally conjugate approach in Algorithm 1 (denoted by random cond, middle boxes), even when using stratification (stratified cond, right boxes) for the sampling, whilst the approximate method (random approx, left boxes, Algorithm 2) is stable and accurate. The average number of observations per split is indicated in parentheses. The reference value is indicated as horizontal line.

5.2 Experiment 2: Comparison of the conditional and approximate method

In this experiment we are interested in predicting on-time arrival of air planes where we use a Bayesian logistic regression model with 17 features (see the appendix for more details and a comparison with another model). There are in total $n = 327,346$ observations. Figure 3 shows the results of our simulations. The reference value for the model evidence has been obtained using importance sampling using a Laplace approximation based on the *maximum a posteriori*. In the current setting the bias is less than -0.5%, even for as many as 50 splits, both for the conditional approach that uses data augmentation (Algorithm 1) and the approximate method based on normal approximations (Algorithm 2). We notice a strong downward bias for the conditional approach, as predicted by Proposition 3. This clearly illustrates a weakness of the conditionally conjugate approach and we observed this behaviour on different data sets (not shown here).

A remedy for the high variance is the use of stratification to construct more homogeneous data shards to improve the performance of the conditional approach. We performed k-means clustering of the features with 10 clusters using the full data set. Then we stratify the observations using the outcome and the cluster membership. A similar approach was used in Zhao and Zhang (2014) to diversify sampling for mini batches in stochastic gradient optimisation. The motivation is to obtain representative samples of the entire data set with every cluster being represented. Although often feasible in practice, this approach goes against the idea of distributed computation as all data has to be seen at once to construct a stratification. The improvement for the conditional approach that comes from stratification is rather limited, as shown in Figure 3.

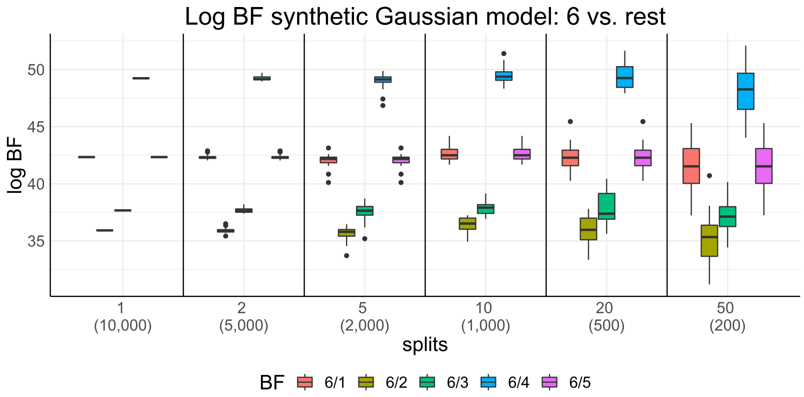


Figure 4: log Bayes factor computed using Algorithm 2 (y-axis) of the correct model 6 against all other models. The BFs stay roughly constant even as the number of splits increase. The average number of observations per split is indicated in parentheses (x-axis). The value for a single split (left most column) serves as reference value. The ordering of the Bayes factors does not change as the number of splits increase and hence consistent model choice is possible over an increasing number of splits.

In the remaining experiments we consider only the approximate method described in Section 4.2 and no stratification.

5.3 Experiment 3: Assessment of the approximate method in a toy example

In this experiment we investigate the behaviour of Algorithm 2 in the same Gaussian toy model as in the first experiment from Section 5.1. We compare the performance for estimating the marginal likelihood over up to 50 splits of $10,000$ observations. The error introduced via the approximation in (10) of the subposterior amounts to less than 0.02% (see Appendix), and despite a small downward bias in estimating $\log p(y)$, the resulting BFs stay stable over decreasing subset sizes (Figure 4), resulting in a consistent choice of the correct model (6). See also in the appendix for the comparison of log marginal likelihoods, that draws a similar picture.

5.4 Experiment 4: The approximate method on a very large data set

This experiment is based on a very large data set from particle physics where a binary classification problem consists in predicting the presence of a Higgs boson. The data set contains 11 million observations. Our aim is to understand whether model (1) with 21 low-level features or model (2) with 7 high-level features is more likely *a posteriori*. Running a full Monte Carlo simulation on the whole data set for model (1) leads to excessive computation times: a full run would take more than 450 hours (almost 3 weeks)

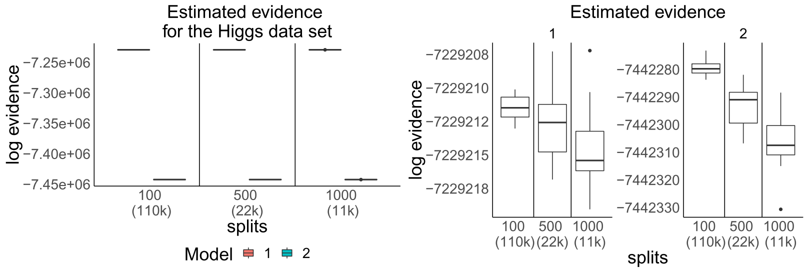


Figure 5: Comparison of the calculated normalising constant using Algorithm 2 (y-axis) for the Higgs data set. The left plot shows the evidence for both models (model 1 left boxes, model 2 right boxes) on the same scale. The middle plot corresponds to model (1), the right plot corresponds to model (2), using their respective scales only. The average number of observations per split is given in parentheses (x-axis). We observe a small downward bias as the number of splits increase.

on a single core CPU (central processing unit). We assess how the model evidence changes as splits get small by dividing the data set in shards of $1\%, 0.2\%$ and 0.1%. Thereby we bring the computation time down to less than 5 hours, 1 hour and less than 30 minutes, all running on different single core CPUs. The combination of the results of the different workers is in the order of a few seconds as we have to perform $\mathcal{O}(Sp^3)$ operations to calculate I_{sub}. (Although necessary matrix inversion can be precomputed locally before sending them to the central node.) We show the results of our estimation in Figure 5. Some bias is introduced by the splitting as the number of shards grow as illustrates the right hand side of Figure 5. However, the bias is overall small and not visible when comparing both models on the same scale (left side of Figure 5). Consequently, we would clearly choose model (1).

In essence our experiment on the Higgs data set illustrates the necessity for distributed computation in the large data regime. Running the same experiment on the entire data set is too slow for most applications.

5.5 Experiment 5: Sparse regression on a large genetic data set

We compare the performance of a linear regression model with a Laplace prior on a real genetic data set from the UK Biobank database. There are $n = 132,353$ observations available. We consider model 1 with 50 and model 2 with 100 genetic variants in the human leukocyte antigen (HLA) region of chromosome 6 that are included as features in order to predict mean red cell volume (MCV). Due to the Laplace prior the conditionally conjugate approach is not applicable. We face again a situation where sampling the posterior on the entire data set on a single core CPU is estimated to take more than 200 hours for model 2. Using 20, 50 and 100 splits brings this computation time down to 10 hours, 2 hours and 1 hour.

In order to assess the bias properly we decided to run our approximate method on a 10% subset of the data where it is computationally feasible to analyse the whole subset

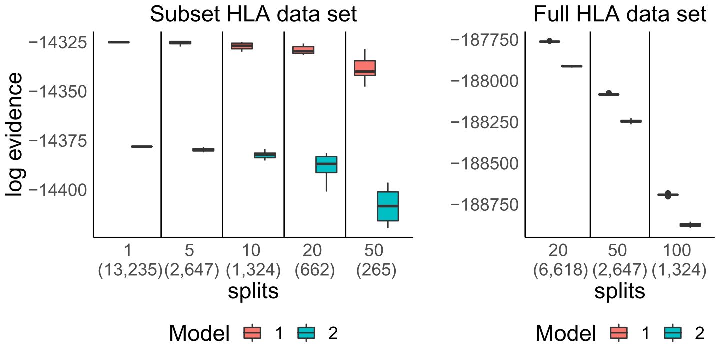


Figure 6: Comparison of the calculated normalising constant using Algorithm 2 (y-axis) for a linear regression using a sparsity enforcing prior. Model 1 (left boxes) has 50 features, model 2 (right boxes) has 100 features. Left plot: estimated model evidence for the model run on a 10% subset. Right plot: estimated model evidence using the full data set but starting with 20 splits. The average number of observations per split is given in parentheses (x-axis). We observe a downward bias as the number of splits increase. For the full data set a comparison across a different number of splits would be meaningless.

and derive a reference value for the normalising constant. We see in Figure 6 on the left side that a downward bias is present, but that we would choose consistently across subsets the right model for the 10% subset of the full data set. When we use the full data set with 20 to 100 splits the right model is still chosen correctly in a given partition (see the right side of the same figure). We also see that model comparison becomes meaningless if different partitions are used to compare the model evidences.

As illustrates Table 1, the error relative to the true value of the normalising constant ($\%\sqrt{\text{MSE}}$) stays small even when using 50 splits and thus having only 265 observations for the estimation of 100 parameters. As the number of splits increases, the squared bias starts to dominate the error as illustrates the ratio Bias^2/Var in Table 1. The error relative to the level of the quantity that we are trying to estimate stays rather small.

SPLITS	5	10	20	50
	MODEL 1 (50 FEATURES)			
$\sqrt{\text{MSE}}$	0.959	2.726	4.537	14.989
$\%\sqrt{\text{MSE}}$	-0.007	-0.019	-0.032	-0.105
$\frac{\text{Bias}^2}{\text{Var}}$	0.04	1.32	3.71	5.44
	MODEL 2 (100 FEATURES)			
$\sqrt{\text{MSE}}$	1.812	4.591	11.404	30.996
$\%\sqrt{\text{MSE}}$	-0.013	-0.032	-0.079	-0.216
$\frac{\text{Bias}^2}{\text{Var}}$	3.42	6.07	2.42	11.99

Table 1: Error metrics of the approximation of the log marginal likelihood for the subset (10%) of the HLA data set with $13,235$ observations.

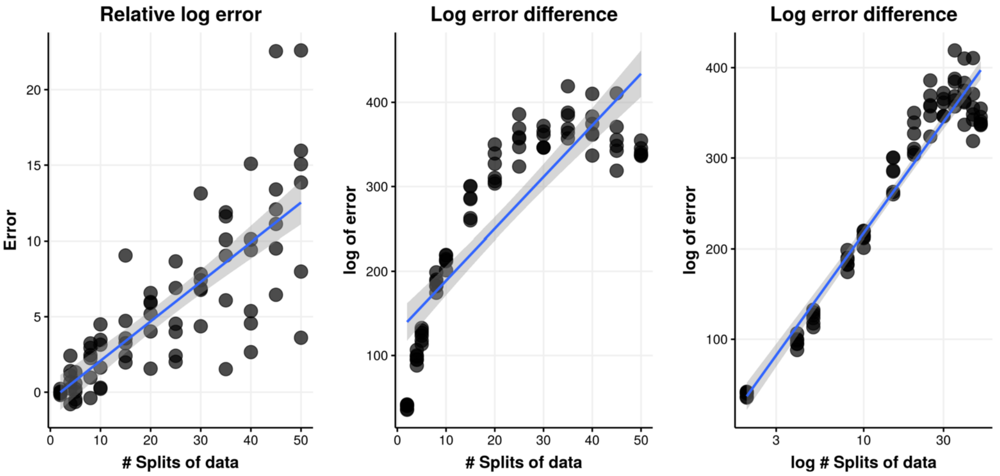


Figure 7: Left plot: error $\epsilon_{1,S}$ as a function of S. Middle plot: error $\epsilon_{2,S}$ as a function of S. Right plot: error $\epsilon_{2,S}$ as a function of $\log S$. The line corresponds to a linear regression line fitted to the data in order to illustrate the trend. We run every simulation 6 times in order to get repeated measurements. The sampler is run over the range of splits $[1, 2, 4, 5, 8, 10, 15, 20, 25, 30, 35, 40, 45, 50]$. The total number of observations is fixed to $13,235$ and varies accordingly with the number of splits.

Finally, for this experiment we assess the error of $\int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s)d\theta$ as the number of splits grows, but the total data set size of $13,235$ stays fixed. We define the following two measures of the error.

$$\epsilon_{1,S} = |\log \int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s)d\theta - \log \int_\Theta \prod_{s=1}^S \mathcal{N}(\theta|\mu_s, \Sigma_s)d\theta|,$$

$$\epsilon_{2,S} = \log |\int_\Theta \prod_{s=1}^S \tilde{p}(\theta|y_s)d\theta - \int_\Theta \prod_{s=1}^S \mathcal{N}(\theta|\mu_s, \Sigma_s)d\theta|.$$

The error $\epsilon_{1,S}$ (referred to as relative log error) corresponds to the log of the error predicted by Proposition 5 and it is expected to grow linearly in S. The error $\epsilon_{2,S}$ (referred to as log error of the difference) is a measure of the absolute error. The left most plot in Figure 7 illustrates that $\epsilon_{1,S}$ seems to grows linearly with S. The error $\epsilon_{2,S}$ seems to grow more like $\log S$, as depicted in the right-most plot in Figure 7.

5.6 Experiment 6: Distributed RJMCMC

Finally, we investigate the use of our splitting approach for a vanilla model selection in a RJMCMC setting. Our simulation should be seen as a proof of concept as RJMCMC faces numerous issues that make exact inference dependent on various tuning parameters that go beyond the scope of this paper.

Variables	1	2	3	4	5
Model 1	✓	✓	✓	×	✓
Model 2	✓	×	×	✓	✓
Model 3	✓	✓	×	×	✓
Truth	✓	✓	1/2	×	✓

Table 2: Active variables in the RJMCMC experiment.

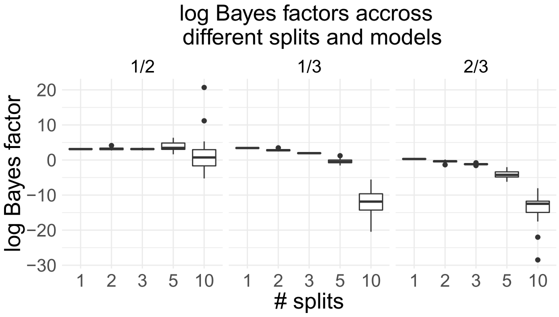


Figure 8: Comparison of the Bayes factor over several splits for the RJMCMC sampler based on Algorithm 3 (y-axis). We compare the BF of model 1 vs 2 (1/2), reference log BF 3.2, model 1 vs 3 (1/3), reference log BF 3.6 and model 2 vs 3 (2/3), reference log BF 0.4 over a changing number of splits (x-axis).

For this purpose we simulate a toy data set of size $n = 4,000$ with a binary outcome where the five features exhibit a high correlation of 0.9.

The data is generated by mixing two data sets. Thus, we artificially generate a setting where it is not clear whether to include the third variable. See Table 2 for more details as well as the appendix. The comparison of the different models is shown in Figure 8. As the number of splits increase, the estimates of the Bayes factors deteriorate. In the current setting going beyond 3 splits may lead to misleading results due to high variance of the estimates as show our experiment. This high variance occurs when combining back the results and is due to several reasons. RJMCMC samplers take a long time to mix and less likely models are potentially not explored enough. Therefore both the estimates based on the MCMC samples of the chain as well as the sojourn times suffer from high variance that accumulates when combining the results from several splits. There is no guarantee that on all data shards all models of interest are explored, if the data shards are too small. In this situation it is not possible to combine the results from several shards back together.

For our experiments we decided to use a medium sized data set and only a small number of features and thereby making exploration of the relevant models more likely. Potential remedies for the evoked problems are an improved estimation of the BF using the method presented in Bartolucci et al. (2006) or the construction of more homogeneous splits using stratification. In any case, whilst theoretically possible, distributed model evidence will rely on well mixing efficient RJMCMC in each split.

5.7 Guidelines for practical application

When using our approach in practice we recommend at least a few thousand observations per data shard for a normal approximation to be reasonable. At this stage we suggest to avoid high dimensional settings where the number of parameters exceeds the number of data points, see also Kass and Raftery (1995), where at least 5 observations per dimension are recommended. If the number of observations per shard are too small, the normal approximation becomes unreliable and one risks to face a large downward bias when combining the results. For complex models the bias grows faster as we split the data in smaller shards. It can be helpful to evaluate the variance of summary statistics across the shards to detect if the splits are not homogeneous and run the sampler on a different number of splits.

We recommend sufficiently long Markov chains (see our default settings for the experiments) and the use of convergence diagnostics to make sure that the posterior has been explored sufficiently and that estimated posterior moments are reliable. In particular, we often face the challenge to find a balance between (a) making sure the approximations are precise enough and (b) limiting computation time. In practice, the bias in the estimation is often smaller than the variation in the estimators. Thus consistent model choice between competing models is possible. However, care is needed if competing models are similar.

6 Discussion and Conclusion

We have presented an approach to calculate the normalising constant in a distributed fashion to enable Bayesian model choice with large data sets. We are able to effectively divide the computation time by several orders of magnitude by splitting the data over a large number of workers and limiting communication between workers. We have shown overall good numerical results and explained the theoretical underpinning for our approach. There remain open questions. Although the estimation of the subposterior normalising constants is biased in general, this bias seems worth accepting in practice. It would be interesting to link this bias of I_{sub} to the way the data is split and to the characteristics of the model such as, e.g., its dimension.

Proposition 4 suggests possible refinements of the normal approximation to I_{sub} (10). In principle, the correction factor

$$\frac{\int_\Theta \prod_{s=1}^S \widetilde{p}(\theta \mid y_s)\, d\theta}{\int_\Theta \prod_{s=1}^S \mathcal{N}(\theta \mid \mu_s, \Sigma_s)\, d\theta} = \mathbb{E}_{\mathcal{N}(\theta|\mu,\Sigma)}\left[\prod_{s=1}^S \frac{\widetilde{p}(\theta \mid y_s)}{\mathcal{N}(\theta|\mu_s,\Sigma_s)}\right] \tag{12}$$

could be estimated by first approximating the density ratio terms $\widetilde{p}(\theta|y_s)/\mathcal{N}(\theta|\mu_s,\Sigma_s)$, and then using a simple Monte Carlo average for the expectation over the global normal approximation $\mathcal{N}(\theta|\mu,\Sigma) \propto \prod_{s=1}^S \mathcal{N}(\theta|\mu_s,\Sigma_s)$ as we suggest for the IS CMC estimator. Subposterior samples could be used to construct a kernel density estimate of $\widetilde{p}(\theta|y_s)$ (Neiswanger et al., 2013). The density ratio could also be estimated using various nonparametric techniques from the machine-learning literature (Kanamori et al., 2012). For large values of S, a Laplace approximation to the expectation could also be accurate enough for practical purposes (Tierney and Kadane, 1986). Estimation of the correction term (12) will introduce some additional variance, however this may be compensated for by a reduction in the bias. A detailed investigation of the costs and benefits of these approaches is an avenue for future research.

If we want to achieve truly parallel Bayesian computation, we must be able to both split the data and run short Markov chains without burn-in bias (Jacob et al., 2020). Based on this idea an unbiased estimation of the normalising constant via bridge sampling (Rischard et al., 2018) could be combined with our method to improve scalability. Another interesting avenue for future research would be the use of variational inference for distributed Bayesian model choice using the work of Rabinovich et al. (2015); Nowozin (2018). In practical settings shotgun stochastic search (SSS) (Hans et al., 2007) could be applicable using our decomposition as SSS relies on normal approximations to the posterior to quickly explore different models. We also suggest to generalise our approach to settings where data subsets are not i.i.d. and of varying seize and investigate applications to federated learning (Li et al., 2020) in combination with posterior approximations for neural networks (Immer et al., 2021). We think that distributed Bayesian computation merits further theoretical and practical investigation as an alternative to the mini batch paradigm.

Supplementary Material

Appendix (DOI: 10.1214/22-BA1321SUPP; .pdf). In the supplementary material we list proofs of our theoretical results, provide details on our experiments and highlight further results.

References

Ahfock, D. C. (2019). “New statistical perspectives on efficient Big Data algorithms for high-dimensional Bayesian regression and model selection.” Ph.D. thesis, University of Cambridge. URL `https://doi.org/10.17863/CAM.38965` 8

Alquier, P., Friel, N., Everitt, R., and Boland, A. (2016). “Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels.” *Statistics and Computing*, 26(1-2): 29–47. MR3439357. doi: `https://doi.org/10.1007/s11222-014-9521-x`. 3

Bardenet, R., Doucet, A., and Holmes, C. (2017). “On Markov chain Monte Carlo methods for tall data.” *The Journal of Machine Learning Research*, 18(1): 1515–1557. MR3670492. 3, 6

Barthelmé, S., Chopin, N., and Cottet, V. (2018). “Divide and conquer in ABC: Expectation-Propagation algorithms for likelihood-free inference.” *Handbook of Approximate Bayesian Computation*, 415–34. MR3889291. 3

Bartolucci, F., Scaccia, L., and Mira, A. (2006). “Efficient Bayes factor estimation from the reversible jump output." *Biometrika*, 93(1): 41–52. MR2277738. doi: https://doi.org/10.1093/biomet/93.1.41. 24

Buchholz, A., Ahfock, D., and Richardson, S. (2022). "Appendix." *Bayesian Analysis*. doi: https://doi.org/10.1214/22-BA1321SUPP. 7, 9, 10, 12, 13, 14, 16

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, Articles*, 76(1): 1–32. URL https://www.jstatsoft.org/v076/i01 15

Chen, H.-Y. and Chao, W.-L. (2021). "FedBE: Making Bayesian Model Ensemble Applicable to Federated Learning." In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=dgtpE6gKjHn 3

Chen, T., Fox, E., and Guestrin, C. (2014). "Stochastic Gradient Hamiltonian Monte Carlo." In Xing, E. P. and Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 1683–1691. Bejing, China: PMLR. URL https://proceedings.mlr.press/v32/cheni14.html 3

Chib, S. (1995). "Marginal Likelihood from the Gibbs Output." *Journal of the American Statistical Association*, 90(432): 1313–1321. MR1379473. 5

Dang, K.-D., Quiroz, M., Kohn, R., Tran, M.-N., and Villani, M. (2019). "Hamiltonian Monte Carlo with Energy Conserving Subsampling." *Journal of Machine Learning Research*, 20(100): 1–31. URL http://jmlr.org/papers/v20/17-452.html MR3990454. 3

Dean, J. and Ghemawat, S. (2008). "MapReduce: simplified data processing on large clusters." *Communications of the ACM*, 51(1): 107–113. 2

Deisenroth, M. and Ng, J. W. (2015). "Distributed Gaussian Processes." In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 1481–1490. Lille, France: PMLR. URL https://proceedings.mlr.press/v37/deisenroth15.html 3

Del Moral, P., Doucet, A., and Jasra, A. (2006). "Sequential Monte Carlo samplers." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3): 411–436. MR2278333. doi: https://doi.org/10.1111/j.1467-9868.2006.00553.x. 4

Dietrich, F. (2010). "Bayesian group belief." *Social Choice and Welfare*, 35(4): 595–626. MR2720740. doi: https://doi.org/10.1007/s00355-010-0453-x. 2, 5

Dümbgen, L., Samworth, R. J., and Wellner, J. A. (2021). "Bounding distributional errors via density ratios." *Bernoulli*, 27(2): 818–852. MR4255217. doi: https://doi.org/10.3150/20-bej1256. 12

Dunson, D. B. and Johndrow, J. E. (2019). "The Hastings algorithm at fifty." *Biometrika*. Asz066. URL https://doi.org/10.1093/biomet/asz066 MR4064137. doi: https://doi.org/10.1093/biomet/asz066. 7

Emerson, P. (2013). "The original Borda count and partial voting." *Social Choice and Welfare*, 40(2): 353–358. URL https://doi.org/10.1007/s00355-011-0603-9 MR3018378. doi: https://doi.org/10.1007/s00355-011-0603-9. 17

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2013). *Bayesian Data Analysis*. Chapman & Hall. Taylor & Francis. MR3235677. 5

Gelman, A. and Meng, X.-L. (1998). "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling." *Statistical Science*, 163–185. MR1647507. doi: https://doi.org/10.1214/ss/1028905934. 4

Gelman, A., Vehtari, A., Jylänki, P., Sivula, T., Tran, D., Sahai, S., Blomstedt, P., Cunningham, J. P., Schiminovich, D., and Robert, C. (2017). "Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data." *arXiv preprint* arXiv:1412.4869. MR4071200. 3

Genest, C. (1984). "A Characterization Theorem for Externally Bayesian Groups." *The Annals of Statistics*, 12(3): 1100–1105. MR0751297. doi: https://doi.org/10.1214/aos/1176346726. 2, 5

Geweke, J. (1989). "Bayesian inference in econometric models using Monte Carlo integration." *Econometrica: Journal of the Econometric Society*, 1317–1339. MR1035115. doi: https://doi.org/10.2307/1913710. 4

Ghosh, J. K. and Ramamoorthi, R. V. (2003). *Bayesian nonparametrics*. Springer Series in Statistics. New York: Springer. URL http://cds.cern.ch/record/1608771 MR1992245. 7, 11

Goudie, R. J., Presanis, A. M., Lunn, D., De Angelis, D., and Wernisch, L. (2019). "Joining and splitting models with Markov melding." *Bayesian Analysis*, 14(1): 81. MR3910039. doi: https://doi.org/10.1214/18-BA1104. 3

Green, P. J. (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination." *Biometrika*, 82(4): 711–732. MR1380810. doi: https://doi.org/10.1093/biomet/82.4.711. 13

Gronau, Q. F., Singmann, H., and Wagenmakers, E.-J. (2020). "bridgesampling: An R Package for Estimating Normalizing Constants." *Journal of Statistical Software*, 92(10): 1–29. URL https://www.jstatsoft.org/index.php/jss/article/view/v092i10 15

Gunawan, D., Dang, K.-D., Quiroz, M., Kohn, R., and Tran, M.-N. (2020). "Subsampling sequential Monte Carlo for static Bayesian models." *Statistics and Computing*, 30(6): 1741–1758. MR4156346. doi: https://doi.org/10.1007/s11222-020-09969-z. 3

Hans, C., Dobra, A., and West, M. (2007). "Shotgun stochastic search for "large p" regression." *Journal of the American Statistical Association*, 102(478): 507–516. MR2370849. doi: https://doi.org/10.1198/016214507000000121. 26

Hastings, W. K. (1970). "Monte Carlo sampling methods using Markov chains and their applications." *Biometrika*, 57(1): 97–109. URL https://doi.org/10.1093/biomet/57.1.97 MR3363437. doi: https://doi.org/10.1093/biomet/57.1.97. 7

Hinton, G. E. (2002). "Training products of experts by minimizing contrastive divergence." *Neural Computation*, 14(8): 1771–1800. 5

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). "Stochastic variational inference." *The Journal of Machine Learning Research*, 14(1): 1303–1347. MR3081926. 3

Holmes, C. C. and Held, L. (2006). "Bayesian auxiliary variable models for binary and multinomial regression." *Bayesian Analysis*, 1(1): 145–168. URL https://doi.org/10.1214/06-BA105 MR2227368. doi: https://doi.org/10.1214/06-BA105. 5

Huang, Z. and Gelman, A. (2005). "Sampling for Bayesian computation with large datasets." *SSRN 1010107*. 6

Immer, A., Bauer, M., Fortuin, V., Rätsch, G., and Emtiyaz, K. M. (2021). "Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning." In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 4563–4573. PMLR. URL https://proceedings.mlr.press/v139/immer21a.html 26

Jacob, P. E., Murray, L. M., Holmes, C. C., and Robert, C. P. (2017). "Better together? Statistical learning in models made of modules." *arXiv preprint* arXiv:1708.08719. 3

Jacob, P. E., O'Leary, J., and Atchadé, Y. F. (2020). "Unbiased Markov chain Monte Carlo methods with couplings." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(3): 543–600. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssb.12336 MR4112777. doi: https://doi.org/10.1111/rssb.12336. 26

Jahan, F., Ullah, I., and Mengersen, K. L. (2020). "A survey of Bayesian statistical approaches for big data." In *Case studies in applied Bayesian data science*, 17–44. Springer. MR4175775. doi: https://doi.org/10.1007/978-3-030-42553-1_2. 3

Jordan, M. I., Lee, J. D., and Yang, Y. (2019). "Communication-efficient distributed statistical inference." *Journal of the American Statistical Association*, 114(526): 668–681. MR3963171. doi: https://doi.org/10.1080/01621459.2018.1429274. 3

Kanamori, T., Suzuki, T., and Sugiyama, M. (2012). "Statistical analysis of kernel-based least-squares density-ratio estimation." *Machine Learning*, 86(3): 335–367. MR2897527. doi: https://doi.org/10.1007/s10994-011-5266-3. 25

Kass, R. E. and Raftery, A. E. (1995). "Bayes factors." *Journal of the American Statistical Association*, 90(430): 773–795. MR3363402. doi: https://doi.org/10.1080/01621459.1995.10476572. 1, 3, 4, 25

Knuth, K. H., Habeck, M., Malakar, N. K., Mubeen, A. M., and Placek, B. (2015). "Bayesian Evidence and Model Selection." *Digitial Signal Processing*, 47(C): 50–67.

URL https://doi.org/10.1016/j.dsp.2015.06.012 MR3425317. doi: https://doi.org/10.1016/j.dsp.2015.06.012. 4

Lewis, S. M. and Raftery, A. E. (1997). "Estimating Bayes factors via posterior simulation with the Laplace—Metropolis estimator." *Journal of the American Statistical Association*, 92(438): 648–655. MR1467855. doi: https://doi.org/10.2307/2965712. 7

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). "Federated Learning: Challenges, Methods, and Future Directions." *IEEE Signal Processing Magazine*, 37(3): 50–60. 3, 26

Lyne, A.-M., Girolami, M., Atchadé, Y., Strathmann, H., and Simpson, D. (2015). "On Russian Roulette Estimates for Bayesian Inference with Doubly-Intractable Likelihoods." *Statistical Science*, 30(4): 443–467. URL https://doi.org/10.1214/15-STS523 MR3432836. doi: https://doi.org/10.1214/15-STS523. 3

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). "Communication-efficient learning of deep networks from decentralized data." In *Artificial intelligence and statistics*, 1273–1282. PMLR. 4

Meng, X.-L. and Wong, W. H. (1996). "Simulating ratios of normalizing constants via a simple identity: a theoretical exploration." *Statistica Sinica*, 831–860. MR1422406. 4

Minsker, S., Srivastava, S., Lin, L., and Dunson, D. (2014). "Scalable and Robust Bayesian Inference via the Median Posterior." In Xing, E. P. and Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 1656–1664. Bejing, China: PMLR. URL http://proceedings.mlr.press/v32/minsker14.html 3

Neal, R. M. et al. (2011). "MCMC using Hamiltonian dynamics." *Handbook of Markov cain Monte Carlo*, 2(11): 2. MR2858447. 7

Neiswanger, W., Wang, C., and Xing, E. (2013). "Asymptotically exact, embarrassingly parallel MCMC." *arXiv preprint* arXiv:1311.4780. 3, 6, 11, 25

Nowozin, S. (2018). "Debiasing Evidence Approximations: On Importance-weighted Autoencoders and Jackknife Variational Inference." In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=HyZoi-WRb 26

Quiroz, M., Kohn, R., Villani, M., and Tran, M.-N. (2019). "Speeding up MCMC by efficient data subsampling." *Journal of the American Statistical Association*, 114(526): 831–843. MR3963184. doi: https://doi.org/10.1080/01621459.2018.1448827. 3

Rabinovich, M., Angelino, E., and Jordan, M. I. (2015). "Variational consensus Monte Carlo." In *Advances in Neural Information Processing Systems*, 1207–1215. 26

Rendell, L. J., Johansen, A. M., Lee, A., and Whiteley, N. (2020). "Global consensus Monte Carlo." *Journal of Computational and Graphical Statistics*, 30(2): 249–259. MR4270501. doi: https://doi.org/10.1080/10618600.2020.1811105. 3

Rischard, M., Jacob, P. E., and Pillai, N. (2018). "Unbiased estimation of log normalizing constants with applications to Bayesian cross-validation." *arXiv preprint* arXiv:1810.01382. 26

Robert, C. (2007). *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media. MR2723361. 1

Safaryan, M., Islamov, R., Qian, X., and Richtárik, P. (2021). "FedNL: Making Newton-Type Methods Applicable to Federated Learning." *CoRR*, abs/2106.02969. URL https://arxiv.org/abs/2106.02969 4

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). "Bayes and big data: The consensus Monte Carlo algorithm." *International Journal of Management Science and Engineering Management*, 11(2): 78–88. 3, 4, 6, 7

Scott, S. L. et al. (2017). "Comparing consensus Monte Carlo strategies for distributed Bayesian computation." *Brazilian Journal of Probability and Statistics*, 31(4): 668–685. MR3738171. doi: https://doi.org/10.1214/17-BJPS365. 3, 7, 11

Skilling, J. (2006). "Nested sampling for general Bayesian computation." *Bayesian Analysis*, 1(4): 833–859. MR2282208. doi: https://doi.org/10.1214/06-BA127. 4

Srivastava, S., Li, C., and Dunson, D. B. (2018). "Scalable Bayes via barycenter in Wasserstein space." *The Journal of Machine Learning Research*, 19(1): 312–346. MR3862415. 3

Szabó, B. and van Zanten, H. (2019). "An asymptotic analysis of distributed nonparametric methods." *Journal of Machine Learning Research*, 20(87): 1–30. MR3960941. 3

Tanner, M. A. and Wong, W. H. (2010). "From EM to Data Augmentation: The Emergence of MCMC Bayesian Computation in the 1980s." *Statistical Science*, 25(4): 506–516. MR2807767. doi: https://doi.org/10.1214/10-STS341. 5

Tierney, L. and Kadane, J. B. (1986). "Accurate Approximations for Posterior Moments and Marginal Densities." *Journal of the American Statistical Association*, 81(393): 82–86. MR0830567. 26

van Dyk, D. A. and Meng, X.-L. (2001). "The Art of Data Augmentation." *Journal of Computational and Graphical Statistics*, 10(1): 1–50. MR1936358. doi: https://doi.org/10.1198/10618600152418584. 5

Wang, X. and Dunson, D. B. (2013). "Parallelizing MCMC via Weierstrass sampler." *arXiv preprint* arXiv:1312.4605. 3

Welling, M. and Teh, Y. W. (2011). "Bayesian Learning via Stochastic Gradient Langevin Dynamics." In Getoor, L. and Scheffer, T. (eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 – July 2, 2011*, 681–688. Omnipress. URL https://icml.cc/2011/papers/398_icmlpaper.pdf MR2867140. doi: https://doi.org/10.1007/978-1-4419-9782-1_28. 3

West, M. (1984). “Bayesian Aggregation.” *Journal of the Royal Statistical Society. Series A (General)*, 147(4): 600–607. MR0782071. doi: https://doi.org/10.2307/2981847. 2

Wickham, H. (2011). “The Split-Apply-Combine Strategy for Data Analysis.” *Journal of Statistical Software*, 40(1): 1–29. URL https://www.jstatsoft.org/index.php/jss/article/view/v040i01 2

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K. H., Hoang, T. N., and Khazaeni, Y. (2019). “Bayesian Nonparametric Federated Learning of Neural Networks.” In *ICML*, 7252–7261. URL http://proceedings.mlr.press/v97/yurochkin19a.html 3

Zhang, M. M., Lam, H., and Lin, L. (2018). “Robust and parallel Bayesian model selection.” *Computational Statistics & Data Analysis*, 127: 229–247. MR3820320. doi: https://doi.org/10.1016/j.csda.2018.05.016. 3, 17

Zhao, P. and Zhang, T. (2014). “Accelerating minibatch stochastic gradient descent using stratified sampling.” *arXiv preprint* arXiv:1405.3080. 19

Zhu, J., Chen, J., Hu, W., and Zhang, B. (2017). “Big learning with Bayesian methods.” *National Science Review*, 4(4): 627–651. 3

Acknowledgments

We thank Leonardo Bottolo, Paul Newcombe, Will Astle and Nicolas Chopin for helpful discussions and Will Astle for providing data. We would like to thank the reviewers and the editor for their feedback that greatly helped to improve our work.