Embarassingly Parallel MCMC using Real NVP

Diego Mesquita [1] **Paul Blomstedt** [1] **Samuel Kaski** [1]

Abstract

Embarrassingly parallel MCMC strategies take a divide-and-conquer stance to scaling up sampling from a posterior by writing the target as a product of subposteriors, running MCMC for each of them in parallel and subsequently combining the results. The challenge then lies in devising efficient aggregation strategies. Our key insight, presented in this extended abstract, is to introduce deep invertible transformations to approximate each of the subposteriors. While current strategies trade-off between accuracy and costs of communication and computation, capitalizing on the properties of these transformations, we are able to circumvent this trade-off.

Markov Chain Monte Carlo (MCMC) algorithms have cemented themselves as a cornerstone of practical Bayesian analysis. Nonetheless, accommodating large distributed datasets is still a challenge. For this purpose, methods have been proposed to speed up inference either using mini-batches (e.g. Ma et al., 2015; Quiroz et al., 2018) or exploiting parallel computing (e.g. Ahn et al., 2014; Johnson et al., 2013), or combinations thereof. For a comprehensive review about scaling up Bayesian inference, we refer to Angelino et al. (2016).

A particularly efficient class of parallel algorithms are embarrassingly parallel MCMC methods, which employ a divide-and-conquer strategy to obtain samples from the posterior

$$p(\theta|\mathcal{D}) \propto p(\theta)p(\mathcal{D}|\theta),$$

where $p(\theta)$ is a prior, $p(\mathcal{D}|\theta)$ is a likelihood function and the data $\mathcal{D}$ are partitioned into K disjoint subsets $\mathcal{D}_1, \ldots, \mathcal{D}_K$. The general idea is to break the global inference into smaller tasks and combine their results, requiring coordination only in the final aggregation stage. More specifically, the target posterior is factorized as

$$p(\theta|\mathcal{D}) \propto \prod_{k=1}^{K} p(\theta)^{1/K} p(\mathcal{D}_k|\theta), \qquad (1)$$

and the right-hand-side factors, referred to as *subposteriors*, are independently sampled from—in parallel—using an MCMC algorithm of choice. The results are then centralized in a coordinating server and aggregated. The core challenge lies in devising strategies which are both accurate and computationally convenient to combine subposterior samples.

Current embarrassingly parallel MCMC strategies trade-off between approximation quality, and costs of communication and computation. In this work, we propose a novel embarrassingly parallel MCMC strategy termed *non-volume-preserving aggregation product* (NAP), which produces accurate, yet cheap samples from the approximate posterior. Our work builds on the insight that subposteriors of arbitrary complexity can be mapped to densities of tractable form using real NVP trasformations (Dinh et al., 2017). This enables us to accurately evaluate the subposterior densities and sample from the combined posterior using importance sampling. After sampling from the subposteriors, we model each of them using a real NVP and combine them using importance sampling. For a sample generated from one of the approximate subposteriors, an importance weight is naturally obtained as the product of the density estimates for the remaining ones.

Experimental results show that NAP outperforms state-of-the art methods in several situations, including heterogeneous subposteriors and intricate-shaped, multi-modal or high-dimensional posteriors. Finally, the proposed strategy results in communication costs which are constant in the number of subposterior samples, which is an appealing feature when communication between machines holding data shards and the server is expensive or limited.

Further details on the method can be found in Mesquita et al. (2019).

EXPERIMENTAL RESULTS

We evaluated the performance of the proposed method in two sets of experiments: on a uni-modal distribution of an

[1] Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University. Correspondence to: Diego Mesquita <diego.mesquita@aalto.fi>.

First workshop on *Invertible Neural Networks and Normalizing Flows* (ICML 2019), Long Beach, CA, USA

intricate shape, and on approximating logistic regression posteriors in moderately high dimensions. We compare our method (NAP) against several available aggregation methods: Consensus Monte Carlo (CON) by Scott et al. (2016); Parallel aggregation using random partition trees (PART) by Wang et al. (2015); and the Parametric (PARAM), Nonparametric (NP) and Semi-parametric (SP) methods proposed by Neiswanger et al. (2014). Implementation details and additional experiments can be found in Mesquita et al. (2019).

All MCMC simulations were carried out using the python interface of the Stan probabilistic programming language (Carpenter et al., 2017). The real NVP networks were implemented with PyTorch[1] using three transformations ($L = 3$) and Gaussian spherical base densities. The scale and translation networks were all implemented as multi-layer perceptrons with two hidden-layers comprising 256 nodes each. The layers of these networks were all equipped with rectified linear units except for the the last layer of each scale network, which was equipped with the hyperbolic tangent activation function. The network parameters were optimized using ADAM (Kingma & Ba, 2014) over 1000 iterations with learning rate 10^{-4}.

WARPED GAUSSIAN

We first consider inference in a warped Gaussian model which exhibits a banana-shaped posterior and is described by the generative model, $y \sim \mathcal{N}(\mu_1 + \mu_2^2, \sigma^2)$, where the true value of the parameters μ_1 and μ_2 are 0.5 and 0, respectively. The variance σ^2 is set to 2 and treated as a known constant. We draw 10000 observations from the model and distribute them in $K = 10$ disjoint sets. Gaussian priors with zero mean and variance 25 were placed both on μ_1 and μ_0.

Figure 1 shows[2] the samples from the approximate posterior obtained with different aggregation methods, plotted against the posterior obtained using the entire sample set. Of all the methods, only NAP and PART were flexible enough to mimic the banana shape of the posterior. PART, however, is overly concentrated when compared to the ground truth, while NAP more faithfully spreads the mass of the distribution.

BAYESIAN LOGISTIC REGRESSION

We now explore how our method behaves in higher dimensions in comparison to its alternatives. For this purpose we consider inference on the simple logistic regression model with likelihood

$$y_i \sim \text{Bernoulli}\big(\sigma(\theta_{1:p} \cdot x_i + \theta_0)\big) \quad \forall 1 \leq i \leq N,$$

where $\cdot$ denotes the dot product, $\sigma(t) = (1 + e^{-t})^{-1}$ is the logistic function, and $\theta_0, \ldots, \theta_p$ receive independent $\mathcal{N}(0, \sqrt{5})$ priors. The true value θ_0' of θ_0 is held at -3 and the remaining $\theta_1', \ldots, \theta_p'$ are independently drawn from a normal distribution with zero-mean and variance 0.25.

To generate a sample pair (x_i, y_i), we first draw x_i from $\mathcal{N}(\mathbf{0}, \Sigma)$, where the covariance matrix Σ is such that

$$\Sigma_{i,j} = 0.9^{|i-j|} \quad \forall 1 \leq i, j \leq p.$$

Then, y_i is computed by rounding $\sigma(\theta_{1:p}' \cdot x_i + \theta_0')$ to one if it is at least 0.5, and to zero otherwise. For each value of $p \in \{25, 50, 100\}$, we draw $N = 10000$ sample pairs using the scheme described above and distribute them in $K = 50$ disjoint sets for parallel inference.

Experiments were repeated ten times for each value of p, in each of which a new θ' was drawn. Table 1 presents the results for each of the aggregation methods in terms of the following performance measures:

- **Root mean squared error** (RMSE) between the mean $\overline{\theta}$ of the approximate posterior samples $\{\theta_r^\star\}_{r=1}^R$ and the mean $\overline{\theta}'$ of samples $\{\theta_r'\}_{r=1}^R$ from the ground truth posterior;
- **Posterior concentration ratio** ($\mathcal{R}$), computed as:

$$\sqrt{\sum_r \|\theta_r - \overline{\theta}'\|_2^2 / \sum_r \|\theta_r' - \overline{\theta}'\|_2^2},$$

  comparing the concentration of the two posteriors around the ground truth mean (values close to one are desirable);
- **KL divergence**; ($\mathrm{D_{KL}}$) between a multivariate normal approximation of the aggregated posterior and a multivariante normal approximation of the true one, both computed from samples.

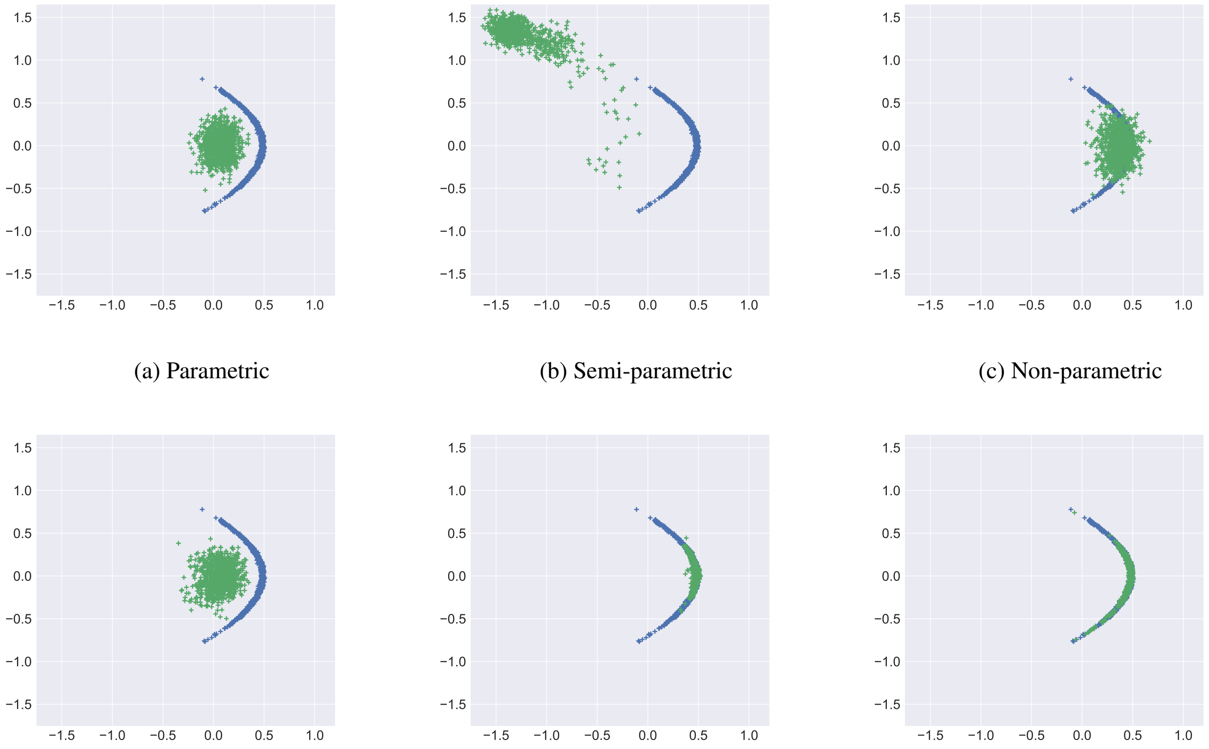
When compared to the other methods, for all values of p, NAP presents a mean closer to the one obtained using centralized inference (smaller RMSE) and has a more accurate spread around it ($\mathcal{R}$ closer to one). In terms of KL divergence, only at $p = 25$, PARAM outperforms NAP by a relatively small margin. Besides this case, NAP performs orders of magnitude better than the other methods, with increasing disparity as p grows.

CONCLUSION

We have proposed an embarrassingly parallel MCMC scheme in which each subposterior density is mapped to

[1] https://pytorch.org

[2] The experiment was repeated with multiple random seeds, yielding similar results.



(a) Parametric

(b) Semi-parametric

(c) Non-parametric

(d) Consensus

(e) PART

(f) NAP

Figure 1. MCMC samples for the warped Gaussian model obtained on the centralized dataset (ground truth), in blue, against samples from posterior approximations using different embarassingly parallel MCMC methods, in green.

	$p = 25$			$p = 50$			$p = 100$		
	RMSE	$\mathcal{R}$	$\mathrm{D_{KL}}$	RMSE	$\mathcal{R}$	$\mathrm{D_{KL}}$	RMSE	$\mathcal{R}$	$\mathrm{D_{KL}}$
NAP	**1.95**	**13.95**	791.86	**1.07**	**13.08**	**1539.32**	**0.63**	**12.43**	**3493.35**
PART	3.29	24.38	4263.53	2.44	31.27	20159.64	1.51	31.80	75423.10
PARAM	2.56	18.34	**589.12**	1.99	24.37	2568.57	1.32	26.07	11245.58
SP	2.43	17.36	1586.80	2.02	24.62	7589.26	1.39	27.26	36994.45
NP	2.39	17.07	1343.88	2.01	24.54	7202.50	1.39	27.26	35313.77
CON	3.51	25.43	10654.78	3.08	37.92	56001.25	2.02	39.96	186275.86

Table 1. Comparison of different aggregation methods for embarassingly parallel MCMC inference on the logistic regression model with p covariates. The values presented are averages over ten repetitions of the experiments. The best results are in bold.

a tractable form using a deep invertible generative model. We capitalized on the ease of sampling from the subposteriors mapped to Real NVP networks and evaluating their respective log density values to build an efficient importance sampling scheme to merge the subposteriors.

It can be shown that, under mild assumptions, our importance sampling scheme is stable, i.e., estimates for a test function h have finite variance (Mesquita et al., 2019). While in this work we gave special attention to the use of real NVP networks, our approach could potentially employ other invertible models, such as the Glow transform (Kingma & Dhariwal, 2018), without loosing theoretical guarantees, as long as the log densities remain bounded.

Our experimental results demonstrated that NAP is capable of capturing intricate posteriors. Moreover, we observed that it significantly outperformed current methods in moderately high-dimensional settings. A possible explanation for this is that, unlike the density estimation techniques underlying the competing methods, the real NVP transformations used in our method are specifically designed for high-dimensional data.

Finally, the generative models we use serve as a intermediate representation of the subposterior, the size of which does not depend on the number of subposterior samples. Thus, workers can produce arbitrarily accurate subposterior estimates by drawing additional samples, without affecting the cost of communicating the subposteriors to the server, or the computational cost of aggregating them into a final posterior estimate.

Acknowledgements

The work was funded by the Academy of Finland, grants 319264 and 294238. The authors gratefully acknowledge the computational resources provided by the Aalto Science-IT project and support from the Finnish Center for Artificial Intelligence FCAI.

References

Ahn, S., Shahbaba, B., and Welling, M. Distributed stochastic gradient MCMC. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, ICML'14, pp. II–1044–II–1052. JMLR.org, 2014.

Angelino, E., Johnson, M. J., and Adams, R. P. Patterns of scalable Bayesian inference. *Foundations and Trends in Machine Learning*, 9(2-3):119–247, 2016. doi: 10.1561/2200000052.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1), 2017.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017.

Johnson, M., Saunderson, J., and Willsky, A. Analyzing hogwild parallel Gaussian Gibbs sampling. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 2715–2723. Curran Associates, Inc., 2013.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Kingma, D. P. and Dhariwal, P. Glow: Generative Flow with Invertible 1x1 Convolutions. *arXiv e-prints*, art. arXiv:1807.03039, Jul 2018.

Ma, Y.-A., Chen, T., and Fox, E. B. A complete recipe for stochastic gradient MCMC. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pp. 2917–2925, Cambridge, MA, USA, 2015. MIT Press.

Mesquita, D., Blomstedt, P., and Kaski, S. Embarrassingly parallel MCMC using deep invertible transformations. *arXiv e-prints*, art. arXiv:1903.04556, Mar 2019.

Neiswanger, W., Wang, C., and Xing, E. P. Asymptotically exact, embarrassingly parallel MCMC. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pp. 623–632, Arlington, Virginia, United States, 2014. AUAI Press.

Quiroz, M., Kohn, R., Villani, M., and Tran, M.-N. Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*, 2018. doi: 10.1080/01621459.2018.1448827.

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

Wang, X., Guo, F., Heller, K. A., and Dunson, D. B. Parallelizing MCMC with random partition trees. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pp. 451–459, Cambridge, MA, USA, 2015. MIT Press.