Mini-Project Outline: Massively Parallel Probabilistic Inference

Sam Bowyer

February 3, 2023

1 Introduction

Importance weighting has found great success in improving probabilistic inference techniques by reweighting several samples at a time drawn from a proposal relevant to the problem at hand. Two well known examples of this are reweighted wake-sleep (RWS) [2] and importance weighted autoencoders (IWAEs) [3], which improve upon wake-sleep [10] and variational autoencoders [11] respectively by calculating objective functions with $K > 1$ weighted samples per latent variable.

However, recent work by Chatterjee and Diaconis [6] suggests that as the number of latent variables n increases, the number of samples needed for effective importance weighting grows with $\mathcal{O}(e^n)$, which can easily render even slightly large importance weighted inference problems intractable. The main idea behind this project is to overcome this issue by drawing K samples per latent variable and then considering all possible K^n combinations of samples. The difficulties involved with working on such a large number of combinations can be ameliorated by considering the dependency graph of the model and using tensor operations (with e.g. PyTorch [13]) on parallelised on GPUs.

This approach was taken in the development of Tensor Monte Carlo (TMC) [1], which showed superior performance to IWAE with only a slight increase in computation time. Ongoing work indicates that a similar approach can be used to improve upon IWAE and RWS further [9], as well as leading to an improved method for computing moments of the latent variables [8]. This variant of TMC is referred to as a "massively parallel" approach and forms the basis of this mini-project.

2 The Massively Parallel Setting

Although it is not feasible to fully derive the massively parallel setting in this document, here we provide a brief introduction to massively parallel importance weights which should give some intuition towards the general approach.

In Bayesian inference we aim to compute a posterior distribution

$$P(z'|x) = \frac{P(x|z')P(z')}{\sum_{z''} P(x, z'')}$$

given a prior $P(z')$ over latent variables z' and a likelihood $P(x|z')$ for data x. However, this computation is often intractable and so instead we work with a proposal distribution $Q(z)$.

Regular importance samples work by drawing K samples from the full joint state space denoted $z \in \mathcal{Z}^K$, with a single sample denoted $z^k \in \mathcal{Z}$. This is obtained by sampling K times from the proposal where $Q(z) = \prod_{k=1}^{K} Q(z^k)$.

We define a 'global' estimator $\mathcal{P}_{\text{global}}(z)$ of the marginal likelihood that can be used in regular IWAE and RWS as follows:

$$\mathcal{P}_{\text{global}}(z) = \tfrac{1}{K} \sum_{k \in \mathcal{K}} r_k(z)$$

where $\mathcal{K} = \{1, ..., K\}$ and

$$r_k(z) = \frac{P(x, z^k)}{Q(z^k|x)}.$$

The idea in the massively parallel approach is to instead compute this estimator over all possible K^n combinations of samples. To show this, we define a vector of indices $\mathbf{k} = (k_1, ..., k_n) \in \mathcal{K}^n$ and write a combination of the samples with these indices as

$$z^{\mathbf{k}} = \left(z_1^{k_1}, z_2^{k_2}, \ldots, z_n^{k_n}\right) \in \mathcal{Z}$$

where $z_i^{k_j}$ represents the k_jth sample of the ith latent variable. With this we can then write the estimator of the marginal likelihood as:

$$\mathcal{P}_{\text{MP}}(z) = \tfrac{1}{K^n} \sum_{\mathbf{k} \in \mathcal{K}^n} r_{\mathbf{k}}(z)$$

where

$$r_{\mathbf{k}}(z) = \frac{P(x, z^{\mathbf{k}})}{\prod_i Q(z_i^k | z_j^k \text{ for } j \in \text{qa}(i))}$$

and $\text{qa}(i)$ gives the parents of the ith latent variable under the proposal. We may then proceed with IWAE and RWS in a massively parallel setting (albeit with further alterations to make computation tractable).

3 Aims

The massively parallel setting has a very large scope as it should be applicable to most probabilistic inference and sampling tasks, hopefully resulting in more efficient methods that are able to effectively use parallel GPU computing. Because of this, a longer-term goal might is to develop a probabilistic programming language (similar to STAN [5] or Turing [7]) with a massively parallel-based implementation. However, the main aims specifically of this mini-project are three-fold:

- To obtain a good understanding of the massively parallel approach and related literature.
- To work on the codebase of the probabilistic programming language. This may include adding testing and documentation which would be useful for the eventual deployment of the language.
- To develop an idea of how massively parallel MCMC methods might be implemented. This will require familiarisation with existing techniques for the parallelisation of MCMC methods, e.g. the approach for Metropolis-Hastings algorithms suggested by Calderhead [4] and Hamiltonian adaptive importance sampling [12].

References

[1] Laurence Aitchison. Tensor Monte Carlo: particle methods for the GPU era, January 2019. arXiv:1806.08593 [cs, stat].

[2] Jörg Bornschein and Yoshua Bengio. Reweighted Wake-Sleep, April 2015. arXiv:1406.2751 [cs].

[3] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance Weighted Autoencoders, November 2016. arXiv:1509.00519 [cs, stat].

[4] Ben Calderhead. A general construction for parallelizing MetropolisHastings algorithms. *Proceedings of the National Academy of Sciences*, 111(49):17408–17413, December 2014. Publisher: Proceedings of the National Academy of Sciences.

[5] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017. Publisher: Columbia Univ., New York, NY (United States); Harvard Univ., Cambridge, MA (United States).

[6] Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2), April 2018.

[7] Hong Ge, Kai Xu, and Zoubin Ghahramani. Turing: A Language for Flexible Probabilistic Inference. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 1682–1690. PMLR, March 2018. ISSN: 2640-3498.

[8] Thomas Heap and Laurence Aitchison. Bayesian inference with massively parallel importance weighting. [Ongoing work].

[9] Thomas Heap and Laurence Aitchison. Massively parallel IWAE and RWS. [Ongoing work].

[10] Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and Radford M. Neal. The "Wake-Sleep" Algorithm for Unsupervised Neural Networks. *Science*, 268(5214):1158–1161, May 1995.

[11] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, 2014. arXiv:1312.6114 [cs, stat].

[12] Ali Mousavi, Reza Monsefi, and Víctor Elvira. Hamiltonian Adaptive Importance Sampling. *IEEE Signal Processing Letters*, 28:713–717, 2021. arXiv:2209.13716 [cs, stat].

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.