Automated Inference Optimizations in Probabilistic Programming Languages

PHD FORUM SUBMISSION

1st Gizem Caylak
EECS, KTH Royal Institute of Technology
Stockholm, Sweden
caylak@kth.se

***Abstract*—Probabilistic programming languages (PPLs) enable users to write a statistical model and leave the inference part to the compile and runtime systems. Ideally, PPL developers aim to separate the inference from writing models, enabling users to focus only on the model where their expertise lies. However, how a user writes a model affects inference efficiency. Further, as the expressiveness of a PPL increases, it becomes harder to implement an efficient inference algorithm. The first part of my PhD thesis focuses on optimization methods that automatically utilize model structures to improve inference efficiency. The main focus of the second part is inference algorithms. The planned contribution includes a comprehensive survey and implementation of various inference algorithms. Based on that survey, we plan to develop efficient inference algorithms for phylogenetics problems.**

***Index Terms*—bayesian inference, probabilistic programming, optimization, belief propagation, delayed sampling**

I. Introduction

The probabilistic programming field has significantly improved model specification and automated Bayesian inference. However, the trade-off between model expressiveness and computational efficiency is a significant challenge, constituting the base of research problems in my PhD thesis.

In Bayesian statistics, we are interested in determining the posterior distribution of model parameters given observed data. According to Bayes' theorem, we can calculate the posterior given the prior distribution of the model parameters and the likelihood of the data. PPLs allow users to define the priors and the likelihoods for models using a high-level programming interface and leave the posterior calculation to the inference algorithms. However, there is a trade-off between the expressiveness of a PPL and inference efficiency. As models grow in complexity, the computational cost of inference increases.

The main research problem of my thesis is *to develop optimization methods to improve inference efficiency in PPLs without compromising model expressiveness.* With this research problem, we aim to find the balance between two factors of probabilistic programming: the ability to write complex models (expressiveness) and the accuracy and execution time of inference algorithms (efficiency).

II. Previous Work

Utilizing structures in a model has been a key component for optimization methods to improve inference efficiency. The analytical relations between random variables in a model may provide closed-form solutions for the inference, improving the inference efficiency [1]. Delayed sampling [2] is a runtime optimization method utilizing conjugate-prior relations between random variables of a model. Baudart et al. [3] and Atkinson et al. [4] propose a modified version of the delayed sampling algorithm for synchronous languages. Atkinson et al. [5] and Azizian et al. [6] improve the delayed sampling algorithm to catch more analytical relations. However, their target language is not universal PPLs. Delayed sampling is a powerful approach; however, the cost of keeping these relations at runtime makes compile-time approaches more appealing.

Lai et al. [7] proposes a delayed sampling approach utilizing these relations at compile-time. However, their approach is not separated from the inference algorithm, meaning that the inference algorithm needs to be adjusted based on their method. Currently, their approach does not support transforming a model, such as latent Dirichlet allocation [8], that contains stochastic branches or unbounded loops. There are other approaches that utilize analytical relations at compile time; however, they either depend on computer algebra [9], or target a non-universal simple PPL [10] [11], or depend on domain-specific languages [12].

In many problem domains, models with tree-structured dependencies among the random variables have been developed to understand hierarchical and evolutionary relationships in real-world data [13, 14, 15, 16, 17] and utilizing the topology of a model, such as random variables forming tree structures, may improve inference efficiency. In Monte Carlo (MC) inference methods, calculating the tree's likelihood is fundamental to estimating the posterior [18, 19]. Naively applying MC methods to the tree-structured models can be computationally expensive since the inference algorithm needs to consider all possible values that an unobserved node can take to calculate the likelihood of a tree. Belief propagation [20] is a method giving exact solutions for such models; however, its integration into universal PPLs requires addressing various challenges to preserve expressiveness and runtime performance.

III. Research Plan

A. What has been done

The thesis has three main contributions that have been completed. The first contribution is *to develop a compile-*

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

time version of delayed sampling that is orthogonal to the inference engine in a universal PPL. Our proposed approach employs the delayed sampling algorithm at compile time and is implemented in a statically typed universal probabilistic programming language, Miking CorePPL [21, 22]. The main idea behind the proposed method is to generate a graph representation of the probabilistic program, transform the graph based on the conjugate prior relations, and reconstruct the program from the transformed graph. The key point here is that we cannot directly represent every structure of a model written in a universal PPL with a Bayesian network (BN) since the program may contain recursion, loops, and stochastic branches. Therefore, we create a graph called a programmatic Bayesian network (PBN) that encapsulates the structures a BN cannot represent directly. In addition to random variable nodes, as in BNs, PBNs contain special nodes such as code blocks, multiplexers, plates, and list nodes to represent a probabilistic program. While PBNs make our approach more expressive, transforming a PBN back to a probabilistic program makes the optimization technique orthogonal to the inference algorithm. We evaluate our contribution on real-world examples, such as latent Dirichlet allocation [8] and demonstrate our contribution's execution time improvement.

The second contribution is *to employ runtime delayed sampling algorithm in a statically-typed universal PPL.* The main idea behind delayed sampling is to delay sampling random variables or not to sample at all, if possible, using the conjugate prior relations. However, if we employ this method in a statically typed system, it introduces a challenge. The type system expects a value type, such as float or integer, for a random variable because of immediate sampling; however, with delayed sampling, depending on whether the value is delayed at runtime, the type can be either a delayed type or a value type. However, the compiler should know the types in a statically typed system. The original delayed sampling algorithms have been implemented in Birch [23] and Anglican [24]. Since Birch is a language designed for delayed sampling, the language constructs account for delayed variables. The Anglican implementation creates separate distributions and constructs for delayed variables. Changing the distributions and constructs to account for delayed variables requires lots of effort for the existing framework in a statically typed system not designed for delayed sampling. To tackle this challenge, we propose a user-annotated system that enables users to mark the delayed random variables, which helps the type system in a statically typed PPL. We evaluate our contribution on real-world examples, such as vector-borne disease model [25].

The third contribution is *to automate the forward pass of belief propagation to improve inference efficiency by utilizing tree structures in a model while preserving the expressiveness.* The forward pass of belief propagation enables us to calculate the likelihood of a tree efficiently. However, its integration into universal PPLs is challenging since the tree structure may not be fixed because of recursion and branches. We need to update the likelihood of the program after observing each subtree. However, not knowing when a subtree is constructed is a challenge. Further, as in delayed sampling, we also encounter type issues because belief propagation does not sample the internal nodes of a tree. We introduce special constructs and a user-annotation system to help the type system. Our approach automatically captures tree structures in a model and applies belief propagation in universal PPLs. We evaluate our approach through case studies in phylogenetics and demonstrate our contribution's inference accuracy using marginalized likelihood variance as a metric. We included all these contributions in two papers, which are under submission.

B. The planned work

The previous contributions consist of utilizing model structures to improve inference efficiency. Although optimization before running inference may improve inference efficiency significantly, choosing or designing an appropriate inference algorithm itself is equally important. For example, in phylogenetics, choosing the appropriate proposal for the inference algorithms is critical for the inference problem to converge [26]. Thus, the next research direction is to survey existing inference algorithms on varying benchmarks. While our focus is on Monte Carlo (MC) methods, such as sequential Monte Carlo [18] and Markov Chain Monte Carlo [19], we plan to provide an analysis of variational inference methods that consider inference as an optimization problem [27]. We plan to implement varying Monte Carlo inference algorithms in Miking CorePPL and determine potential improvements. The main reason for not focusing on variational inference is that phylogenetic studies that we plan to improve the inference on for further studies focus on MC methods [28, 29]. The final planned contribution is to enhance these algorithms to efficiently handle the inference problems in the phylogenetic field. We aim to have these contributions

- A comprehensive analysis of inference algorithms in the context of probabilistic programming and phylogenetic field.
- Implementing a selection of the inference algorithms in Miking CorePPL. We will evaluate their performance on varying models to discuss which inference algorithms are better for certain models and identify potential improvements.
- Enhancement of the existing inference algorithms regarding execution time and inference accuracy based on the analysis from our survey for phylogenetic models.
- Evaluation of the modified inference algorithms on a series of case studies in phylogenetics.

IV. Conclusion

With this research, we aim to improve inference efficiency in universal PPLs. Utilizing model structures, as shown in our previous works, provides significant improvements in inference efficiency while preserving the expressiveness of the models. The next research direction includes a comprehensive survey of inference algorithms on varying models and the implementation of selected inference algorithms in Miking CorePPL. The survey will provide a guide for choosing suitable inference algorithms based on the model properties and insights to improve inference for phylogenetic models.

References

[1] D. Blackwell, “Conditional Expectation and Unbiased Sequential Estimation,” *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 105 – 110, 1947. [Online]. Available: https://doi.org/10.1214/aoms/1177730497

[2] L. Murray, D. Lunde;n, J. Kudlicka, D. Broman, and T. B. Schön, “Delayed sampling and automatic rao-blackwellization of probabilistic programs,” in *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS), Lanzarote, Spain, April, 2018 :*, ser. Proceedings of Machine Learning Research, vol. 84, 2018. [Online]. Available: http://proceedings.mlr.press/v84/murray18a/murray18a.pdf

[3] G. Baudart, L. Mandel, E. Atkinson, B. Sherman, M. Pouzet, and M. Carbin, “Reactive probabilistic programming,” in *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 898–912.

[4] E. Atkinson, G. Baudart, L. Mandel, C. Yuan, and M. Carbin, “Statically bounded-memory delayed sampling for probabilistic streams,” *Proc. ACM Program. Lang.*, vol. 5, no. OOPSLA, oct 2021. [Online]. Available: https://doi.org/10.1145/3485492

[5] E. Atkinson, C. Yuan, G. Baudart, L. Mandel, and M. Carbin, “Semi-symbolic inference for efficient streaming probabilistic programming,” *Proc. ACM Program. Lang.*, vol. 6, no. OOPSLA2, oct 2022. [Online]. Available: https://doi.org/10.1145/3563347

[6] W. Azizian, G. Baudart, and M. Lelarge, “Automatic rao-blackwellization for sequential monte carlo with belief propagation,” in *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. [Online]. Available: https://openreview.net/forum?id=YNf2XCQqM1

[7] J. Lai, J. Burroni, H. Guan, and D. Sheldon, “Automatically marginalized MCMC in probabilistic programming,” in *Fifth Symposium on Advances in Approximate Bayesian Inference*, 2023. [Online]. Available: https://openreview.net/forum?id=lmLRNZU0MY

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[9] C.-c. Shan and N. Ramsey, “Exact Bayesian inference by symbolic disintegration,” in *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, ser. POPL 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 130–144. [Online]. Available: https://doi.org/10.1145/3009837.3009852

[10] D. Huang, J.-B. Tristan, and G. Morrisett, “Compiling markov chain monte carlo algorithms for probabilistic modeling,” in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 111–125. [Online]. Available: https://doi.org/10.1145/3062341.3062375

[11] A. V. Nori, C.-K. Hur, S. K. Rajamani, and S. Samuel, “R2: An efficient mcmc sampler for probabilistic programs,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI’14. AAAI Press, 2014, p. 2476–2482.

[12] M. D. Hoffman, M. J. Johnson, and D. Tran, “Autoconj: Recognizing and exploiting conjugacy without a domain-specific language,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[13] R. D. Gray and Q. D. Atkinson, “Language-tree divergence times support the anatolian theory of indo-european origin,” *Nature*, vol. 426, no. 6965, p. 435—439, November 2003.

[14] P. Kapli, Z. Yang, and M. J. Telford, “Phylogenetic tree building in the genomic age,” *Nature reviews. Genetics*, vol. 21, no. 7, p. 428—444, July 2020.

[15] A. Clauset, C. Moore, and M. E. J. Newman, “Hierarchical structure and the prediction of missing links in networks,” *Nature*, vol. 453, no. 7191, p. 98—101, May 2008.

[16] A. Kassian, “Towards a formal genealogical classification of the lezgian languages (north caucasus): Testing various phylogenetic methods on lexical data,” *PLOS ONE*, vol. 10, no. 2, pp. 1–25, 02 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0116950

[17] M. Dunn, S. J. Greenhill, S. C. Levinson, and R. D. Gray, “Evolved structure of language shows lineage-specific trends in word-order universals,” *Nature*, vol. 473, no. 7345, p. 79—82, May 2011.

[18] A. Doucet, N. De Freitas, N. J. Gordon *et al.*, *Sequential Monte Carlo methods in practice*. Springer, 2001, vol. 1, no. 2.

[19] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, “An introduction to MCMC for machine learning,” *Machine Learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[20] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[21] D. Broman, “A vision of Miking: Interactive programmatic modeling, sound language composition, and self-learning compilation,” in *Proceedings of the 12th ACM SIGPLAN International Conference on Software Language Engineering*. Association for Computing Machinery, 2019, pp. 55–60.

[22] D. Lundén, J. Öhman, J. Kudlicka, V. Senderov, F. Ronquist, and D. Broman, “Compiling universal probabilistic programming languages with efficient parallel sequential monte carlo inference,” in *Programming Languages and Sleystems*. Cham: Springer International Publishing, 2022, pp. 29–56.

[23] L. M. Murray and T. B. Schön, “Automated learning with a probabilistic programming language: Birch,” *Annual Reviews in Control*, vol. 46, pp. 29–43, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578818301202

[24] D. Lundén, “Delayed sampling in the probabilistic programming language anglican,” 2017.

[25] S. Funk, A. J. Kucharski, A. Camacho, R. M. Eggo, L. Yakob, L. M. Murray, and W. J. Edmunds, “Comparative analysis of dengue and zika outbreaks reveals differences by setting and virus,” *PLOS Neglected Tropical Diseases*, vol. 10, no. 12, pp. 1–16, 12 2016. [Online]. Available: https://doi.org/10.1371/journal.pntd.0005173

[26] C. Lakner, P. van der Mark, J. P. Huelsenbeck, B. Larget, and F. Ronquist, “Efficiency of Markov Chain Monte Carlo Tree Proposals in Bayesian Phylogenetics,” *Systematic Biology*, vol. 57, no. 1, pp. 86–103, 02 2008. [Online]. Available: https://doi.org/10.1080/10635150801886156

[27] A. K. David M. Blei and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017. [Online]. Available: https://doi.org/10.1080/01621459.2017.1285773

[28] J. P. Huelsenbeck, F. Ronquist, R. Nielsen, and J. P. Bollback, “Bayesian inference of phylogeny and its impact on evolutionary biology,” *Science*, vol. 294, no. 5550, pp. 2310–2314, 2001. [Online]. Available: http://www.jstor.org/stable/3085235

[29] F. Ronquist, J. Kudlicka, V. Senderov, J. Borgström, N. Lartillot, D. Lundén, L. Murray, T. B. Schön, and D. Broman, “Universal probabilistic programming offers a powerful approach to statistical phylogenetics,” *Communications Biology*, vol. 4, no. 1244, 2021. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:nrm:diva-4601