juliacon

ReactiveMP.jl: A Julia Package for Reactive Message Passing-based Bayesian Inference

Dmitry Bagaev[1] and Bert de Vries[1]

[1]Electrical Eng. Dept., Eindhoven Univ. of Technology, Eindhoven, The Netherlands

ABSTRACT

ReactiveMP.jl is a native Julia implementation of reactive message passing-based Bayesian inference in probabilistic graphical models with Factor Graphs. The package does Constrained Bethe Free Energy minimisation and supports both exact and variational Bayesian inference, provides a convenient syntax for model specification and allows for extra factorisation and form constraints specification of variational family of distributions. In addition, ReactiveMP.jl includes a large range of standard probabilistic models and can easily be extended to custom novel nodes and message update rules. In contrast to non-reactive (imperatively coded) Bayesian inference packages, ReactiveMP.jl scales easily to support inference on a standard laptop for large conjugate models with tens of thousands of variables and millions of nodes.

Keywords

Bayesian Inference, Julia, Factor Graphs, Graphical Models, Free Energy Minimisation, Message Passing, Reactive Programing, Variational Inference

1. Background

Bayesian inference is one of the key computational mechanisms that underlies probabilistic model-based machine learning applications. Unfortunately, for many practical models, Bayesian inference requires evaluating high-dimensional integrals that have no analytical solution. As a result, Probabilistic Programming (PP) tools for Automated Approximate Bayesian Inference (AABI) become popular, e.g., *Turing.jl* [3], *ForneyLab.jl* [2] and others. These tools help researchers to specify probabilistic models in a high-level domain-specific language and run AABI algorithms with minimal additional overhead.

2. Message Passing

An important issue in the development of PP frameworks is scalability of AABI algorithms for large models and large data sets. One solution approach concerns message passing-based inference in factor graphs. In this framework, relationships between model variables are represented by a graph of sparsely connected nodes, and inference proceeds efficiently by a sequence of nodes sending probabilistic messages to neighboring nodes. While the optimal message passing schedule is data-dependent, all existing factor graph frameworks (e.g., Infer.Net [4], ForneyLab.jl) use preset message sequence schedules. In our work we exploit reactive programming approach in the context of message passing based Bayesian inference. The potential benefits of reactive message passing in a factor graph include scaling to large inference tasks, much smaller processing latency and processing of data samples that arrive at irregular time intervals.

3. Reactive Message Passing

We present **ReactiveMP.jl** package, which is a native Julia [1] package for automated *reactive* message passing-based Bayesian inference and corresponding Constrained Bethe Free Energy (CFBE) functional optimisation [6]. ReactiveMP.jl is based on a reactive programming approach, does not enforce any particular message-passing schedule, supports real-time data inference. In our experiments new implementation scales comfortably to inference tasks on factor graphs with tens of thousands of variables and millions of nodes.

The package comes with a collection of standard probabilistic models, including linear Gaussian state-space models, hidden Markov models, auto-regressive models and mixture models. Moreover, ReactiveMP.jl API supports various processing modes such as offline learning, filtering of infinite data streams and protocols for handling missing data.

ReactiveMP.jl is customizable and provides an easy way to add new models, node functions and analytical message update rules to the existing platform. As a result, a user can extend built-in functionality with custom nodes to run automated inference in novel probabilistic models. The resulting inference procedures are differentiable with *ForwardDiff.jl* [5]. In addition, the inference engine supports different types of floating point numbers, e.g., the built-in BigFloat Julia type. As for computation time and memory usage, specifically for conjugate models, ReactiveMP.jl outperforms Turing.jl and ForneyLab.jl significantly by orders of magnitude. Performance benchmarks are available at the GitHub repository.

4. Conclusions

Automating scalable Bayesian inference is a key factor in the quest to apply Bayesian machine learning to useful applications. We developed ReactiveMP.jl as a package that enables developers to build large novel probabilistic models and automate scalable inference in those models by reactive message passing in a factor graph.

5. Acknowledgements

We acknowledge contributions from Albert Podusenko, Ismail Senoz, and Bart van Erp, and support from the whole BIASlab group during this project.

6. References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. doi:10.1137/141000671.

[2] Marco Cox, Thijs van de Laar, and Bert de Vries. A factor graph approach to automated design of Bayesian signal processing algorithms. *International Journal of Approximate Reasoning*, 104:185–204, January 2019. doi:10.1016/j.ijar.2018.11.002.

[3] Hong Ge, Kai Xu, and Zoubin Ghahramani. Turing: a language for flexible probabilistic inference. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pages 1682–1690, 2018.

[4] T. Minka, J.M. Winn, J.P. Guiver, Y. Zaykov, D. Fabian, and J. Bronskill. /Infer.NET 0.3, 2018. Microsoft Research Cambridge. http://dotnet.github.io/infer.

[5] J. Revels, M. Lubin, and T. Papamarkou. Forward-mode automatic differentiation in Julia. *arXiv:1607.07892 [cs.MS]*, 2016.

[6] İsmail Şenöz, Thijs van de Laar, Dmitry Bagaev, and Bert de Vries. Variational message passing and local constraint manipulation in factor graphs. *Entropy*, 23(7), 2021. doi:10.3390/e23070807.