

My JuliaCon proceeding

1st author¹, 2nd author^{1, 2}, and 3rd author²

¹University

²National Lab

ABSTRACT

This is a guide for authors who are preparing papers for JuliaCon using the L^AT_EX document preparation system and the `juliacon` class file.

Keywords

Julia, Optimization, Game theory, Compiler

1. Introduction

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) have become popular, e.g., *Turing.jl* [2], *ForneyLab.jl* [1] 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.

We present **ReactiveMP.jl** package, which is a native Julia package for automated *reactive* message passing-based (both exact and approximate) Bayesian inference and Constrained Bethe Free Energy (CFBE) optimisation [4]. New package 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 provides an easy way to add new models, node functions and analytical message update rules to the existing platform. The resulting inference procedures are differentiable with the *ForwardDiff.jl* [3]. As for computation time and memory usage, specifically for conjugate models, the ReactiveMP.jl outperforms *Turing.jl* and *ForneyLab.jl* significantly by orders of magnitude. Performance benchmarks are available at the GitHub repository.

2. Acknowledgements

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

3. References

- [1] 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.
- [2] 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.
- [3] J. Revels, M. Lubin, and T. Papamarkou. Forward-mode automatic differentiation in Julia. *arXiv:1607.07892 [cs.MS]*, 2016.
- [4] 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.