Reviews

|  |  |
| --- | --- |
| **Review 3** | |
| Significance of the Contribution: | 7: (++) |
| Soundness and Positioning with Respect to Related Work: | 3: (---) |
| Depth of Theoretical and/or Experimental Analysis (as appropriate): | 6: (+ (slightly positive)) |
| Quality of Presentation: | 7: (++) |
| SUMMARY RATING: | **2**: (++) |
| Comments for the Authors: | This paper seeks to address the challenge of doing MCMC in graphical models that have deterministic constraints amongst the variables. The key idea in the paper is to define a class of models such that the following algorithm will work to compute a single-site Gibbs sampling update for variable i: 0. Replace all determistically-constructed variables in the model with the expressions of other variables that they are constrained to be equal to. 1. For each remaining variable, symbolically express the conditional pdf P(xi | x\_{-i}).  Solve for xi in terms of other variables. 2. Symbolically integrate the result to get a symbolic conditional CDF for xi. 3. Use CDF inversion (presumably using binary search) to sample xi | x\_{-i}.  The paper then suggests to do steps 0-2 as a preprocessing step, so that only step 3 is required for each Gibbs update. Experiments show this yields around an order of magnitude speedup over a Gibbs sampler that does steps 1 & 2 nonsymbolically but at every iteration of the sampling.  There are some pros and cons to this paper. The main pro is that the approach is reasonable, and the high level idea of doing some symbolic pre-processing to speed up MCMC inference is compelling. The main con is that there is lots of related work that is ignored in the text and not compared to experimentally.   Pros:   - The paper appears technically sound, is well organized, and the motivation is clear.  - The class of models, Piecewise Algebraic Graphical Models, that are  presented along with the sampling method seems like an interesting and powerful class of model.  - The use of symbolic computations to speed up Gibbs sampling for a general class of models is interesting (of course, people already apply this sort of reasoning when building one-off Gibbs samplers for individual problems, when they construct and simplify conditionals with pen-and-paper, then implement the simplified updates).   Cons:   - The treatment of related work is quite simplistic. More advanced methods of MCMC are quite common these days and have made their way into general-purpose toolboxes, e.g., HMC in Stan [B], SMC methods in Anglican [C], and these toolboxes support sampling with deterministic variables. Thus, the discussion in the intro is over-stating the novelty of the paper. It may be the case that the current paper can support models where these approaches would fail, or that it works better, but this discussion is more nuanced than what appears in the paper, and it should be included. Also related is [A] and the citations within.  - Related point: the baselines are too weak to convince me that this is the best approach. Stan and Anglican are readily available to download and should be compared against when possible. At a quick glance, PyMC [3] appears to support at least the Collision example using the Deterministic class within the modelling language.  - There needs to be a precise discussion of the runtimes of the symbolic computations. Please give expressions in Big-O notation of the solving of the symbolic equations and of the symbolic integration. Is there any case in which these costs will grow exponentially? Relatedly, the collapsing determinism can presumably lead to exponential blow-up in the size of the model representation. This should be discussed as well.  - All of the experiments are toy.   In total:  I found the content of the paper to be interesting, and I think with a bit of revision and more nuanced discussion of related work and stronger baselines, this would be a good paper.    Detailed points / questions:  - Please move the legend in fig 4 (c)  - What is the proposal distribution for the rejection samplers?   - I didn't understand the discussion of the inapplicability of rejection sampling in experiment 3. Just because the log pdf is unbounded doesn't mean there isn't some proposal distribution that could serve as an appropriate envelope.  - Are there situations where it is possible to go one step further and symbolically compute an inverse CDF function, so that step 3 above would not require binary search?  - "... source voltage V and observed" -> "... source voltage V are observed" ->   - How is InverseTransformSample implemented? Binary search to invert the CDF?   [A] https://www.cs.berkeley.edu/~russell/papers/aistats13-dysc.pdf [B] http://mc-stan.org/manual.html [C] http://www.robots.ox.ac.uk/~fwood/anglican/ [D] http://pymc-devs.github.io/pymc/ |

|  |  |
| --- | --- |
| **Review 2** | |
| Significance of the Contribution: | 6: (+ (slightly positive)) |
| Soundness and Positioning with Respect to Related Work: | 6: (+ (slightly positive)) |
| Depth of Theoretical and/or Experimental Analysis (as appropriate): | 5: (- (slightly negative)) |
| Quality of Presentation: | 4: (--) |
| SUMMARY RATING: | **-1**: (- (slightly negative)) |
| Comments for the Authors: | The paper presents a new inference scheme called Symbolic Gibbs sampling for hybrid graphical models that have non-linear deterministic constraints. It introduces a class of problems (Polynomial-Piecewise Polynomial Fractions) that facilitates inference in the presence of linear/nonlinear algebraic deterministic constraints.   In my opinion, the most interesting idea in the paper (and I believe is the main contribution of the paper) is marginalizing a variable by symbolically integrating (indefinite integral) it and then evaluating the integral in presence of the constraints.   However, although the idea is interesting, the paper fails to provide important details about the experiments as well as the main algorithm. Ihe authors have failed to formally define many notations used. It seems like the main inspiration behind this work is limitation of the BUGS system (I am not sure how often in practice non-linear constraints appear).  Minor points and questions:  1> In step 3, Algorithm 1 (Joint factor formulation): The algorithm requires to multiply over all the factors and create a joint factor. This step can itself be intractable since the size of the joint factor will be exponential in the number of variables. An alternate (better?) way might be to use the Shenoy-Shafer architecture. 2> The Dirac delta has been used after equation (1) without mentioning that delta is Dirac delta. 3> The U(\*, \*) distribution is not defined. Is it a uniform distribution (I am assuming it is)?  4> What are the error threshold measures used in Experiment 3 and 4? 5> How the error is computed for Experiment 3, and 4? (i.e.- how the exact answer is computed?) 7> In the step 4 of Algorithm 1 how is the SOLVE method implemented (did you use any symbolic equation solver, or is it hand computed for the example problems?)  I am curious to know how the sampling algorithm performs if these hard constraints are converted to some form of penalty distribution (which heavily penalizes samples that violate the constraint). For example instead of using (hard) Dirac function, may be a softer penalty function (like a gaussian penalty, e.g: N(-(x-G^x)^2, \epsilon) for constraint x=G^x) could have been used. This might have helped the collapsing steps. |

|  |  |
| --- | --- |
| **Review 1** | |
| Significance of the Contribution: | 8: (+++) |
| Soundness and Positioning with Respect to Related Work: | 8: (+++) |
| Depth of Theoretical and/or Experimental Analysis (as appropriate): | 7: (++) |
| Quality of Presentation: | 8: (+++) |
| SUMMARY RATING: | **4**: (++++) |
| Comments for the Authors: | This paper addresses the problem of handling deterministic constraints between random variables in approximate inference strategies like Gibbs sampling. The main idea of the symbolic Gibbs sampling method introduced is to represent possibly nonlinear deterministic rules as polynomial fractions, and to overcome computational problems by analytically pre-computing the univariate cdfs required in Gibbs sampling. In my opinion this work addresses a problem that is both relevant and challenging, where relevance stems from the fact that deterministic relations between random variables are common in many real-world inference scenarios, and challenge refers to the inherent difficulties that arise in traditional MCMC methods with mixed stochastic-deterministic relations.  The paper is written in a clear and transparent way. Both the introductory and the technical parts are easy to follow. The only (minor) problem I see is the not fully convincing experimental validation: the experiments provided are OK, but for some experiments (particularly exp 3 and 4) I found it non-trivial to follow the exact experimental setup and to estimate the resulting difficulty of the task. |

|  |  |
| --- | --- |
| **Review 3** | |
| Significance of the Contribution: | 7: (++) |
| Soundness and Positioning with Respect to Related Work: | 3: (---) |
| Depth of Theoretical and/or Experimental Analysis (as appropriate): | 5: (- (slightly negative)) |
| Quality of Presentation: | 7: (++) |
| SUMMARY RATING: | **1**: (+ (slightly positive)) |
| Comments for the Authors: | This paper seeks to address the challenge of doing MCMC in graphical models that have deterministic constraints amongst the variables. The key idea in the paper is to define a class of models such that the following algorithm will work to compute a single-site Gibbs sampling update for variable i: 0. Replace all determistically-constructed variables in the model with the expressions of other variables that they are constrained to be equal to. 1. For each remaining variable, symbolically express the conditional pdf P(xi | x\_{-i}).  Solve for xi in terms of other variables. 2. Symbolically integrate the result to get a symbolic conditional CDF for xi. 3. Use CDF inversion (presumably using binary search) to sample xi | x\_{-i}.  The paper then suggests to do steps 0-2 as a preprocessing step, so that only step 3 is required for each Gibbs update. Experiments show this yields around an order of magnitude speedup over a Gibbs sampler that does steps 1 & 2 nonsymbolically but at every iteration of the sampling.  There are some pros and cons to this paper. The main pro is that the approach is reasonable, and the high level idea of doing some symbolic pre-processing to speed up MCMC inference is compelling. The main con is that there is lots of related work that is ignored in the text and not compared to experimentally.   Pros:   - The paper appears technically sound, is well organized, and the motivation is clear.  - The class of models, Piecewise Algebraic Graphical Models, that are  presented along with the sampling method seems like an interesting and powerful class of model.  - The use of symbolic computations to speed up Gibbs sampling for a general class of models is interesting (of course, people already apply this sort of reasoning when building one-off Gibbs samplers for individual problems, when they construct and simplify conditionals with pen-and-paper, then implement the simplified updates).   Cons:   - The treatment of related work is quite simplistic. More advanced methods of MCMC are quite common these days and have made their way into general-purpose toolboxes, e.g., HMC in Stan [B], SMC methods in Anglican [C], and these toolboxes support sampling with deterministic variables. Thus, the discussion in the intro is over-stating the novelty of the paper. It may be the case that the current paper can support models where these approaches would fail, or that it works better, but this discussion is more nuanced than what appears in the paper, and it should be included. Also related is [A] and the citations within.  - Related point: the baselines are too weak to convince me that this is the best approach. Stan and Anglican are readily available to download and should be compared against when possible. At a quick glance, PyMC [3] appears to support at least the Collision example using the Deterministic class within the modelling language.  - There needs to be a precise discussion of the runtimes of the symbolic computations. Please give expressions in Big-O notation of the solving of the symbolic equations and of the symbolic integration. Is there any case in which these costs will grow exponentially? Relatedly, the collapsing determinism can presumably lead to exponential blow-up in the size of the model representation. This should be discussed as well.  - All of the experiments are toy.   In total:  I found the content of the paper to be interesting, and I think with a bit of revision and more nuanced discussion of related work and stronger baselines, this would be a good paper.    Detailed points / questions:  - Please move the legend in fig 4 (c)  - What is the proposal distribution for the rejection samplers?   - I didn't understand the discussion of the inapplicability of rejection sampling in experiment 3. Just because the log pdf is unbounded doesn't mean there isn't some proposal distribution that could serve as an appropriate envelope.  - Are there situations where it is possible to go one step further and symbolically compute an inverse CDF function, so that step 3 above would not require binary search?  - "... source voltage V and observed" -> "... source voltage V are observed" ->   - How is InverseTransformSample implemented? Binary search to invert the CDF?   [A] https://www.cs.berkeley.edu/~russell/papers/aistats13-dysc.pdf [B] http://mc-stan.org/manual.html [C] http://www.robots.ox.ac.uk/~fwood/anglican/ [D] http://pymc-devs.github.io/pymc/ |