Optimizing Markov Random Field Inference via Event-Driven Gibbs Sampling for GPUs

Ramin Bashizade, Xiangyu Zhang, Sayan Mukherjee, Alvin R. Lebeck
Duke University
Durham, NC, USA
ramin@cs.duke.edu, xiangyu.zhang@duke.edu, sayan@stat.duke.edu, alvy@cs.duke.edu

I. Introduction

A Markov Random Field (MRF) is a graphical model for representing a wide range of applications in statistical machine learning [4]. MRF encodes the conditional dependence among random variables (RVs). One approach to solving problems represented by MRF is using probabilistic algorithms, e.g., Gibbs sampling [2]. These methods go through all RVs in the MRF model and update them iteratively, until converged to the final result. This process relies on sampling from probability distributions, which is often computationally intensive.

Despite the challenges, the statistical properties of these algorithms, especially their interpretability, make them an attractive alternative approach to deep learning and in some cases, the only approach for certain problems. Therefore, developing solutions to accelerate these algorithms are of significant importance. To achieve this, we can adopt algorithmic optimizations to avoid performing unnecessary work.

In this work, we build on three observations that reveal when RVs cannot change their labels during the current iteration: i) after the warm-up period, most RVs tend to not change labels very often, ii) an RV can only change its label if either it has a non-concentrated probability distribution function (PDF), i.e., it has non-zero probabilities of taking on multiple labels, or at least one of the RVs on which it is conditionally dependent has changed its label, i.e., its PDF has changed, and iii) approximation techniques make it increasingly likely that RVs have concentrated PDFs. Therefore, we introduce event-driven Gibbs sampling (EDGS). In this scheme, queues are used to keep track of RVs that must be updated. To be more specific, a RV is added to the queue if i) another RV on which it is conditionally dependent changes its value, or ii) it does not have a concentrated PDF. We implement EDGS for GPUs to take advantage of the high amount of parallelism provided by them. Our evaluations show up to 30.3% speedup can be gained for stereo vision.

II. Background and Motivation

A. First-order Markov Random Field

A MRF encodes the conditional dependence among RVs. Figure 1 (right) illustrates an example first-order MRF model and its connection with the Gibbs sampling algorithm. In the model, each RV depends on its four immediate neighbors. Due to this structure, the MRF can be divided into two regions so that all of RVs in each region are conditionally independent. This enables us to generate a chromatic schedule to update all RVs in each region in parallel [2].

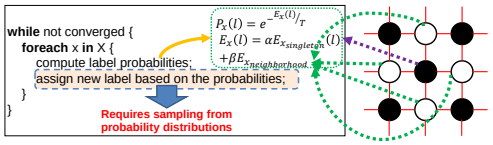


Fig. 1: Markov Chain Monte Carlo algorithm (left) for Markov Random Field (right) inference. Note that sampling is performed in the inner loop.

B. Probabilistic Algorithms

Bayesian inference combines new evidence and prior beliefs to update the probability estimate for a hypothesis. One approach to solving these inference problems is to use probabilistic Markov chain Monte-Carlo (MCMC) methods, e.g., Gibbs sampling [2], that converge to an exact solution by iteratively generating samples for RVs. Figure 1 shows this process. Gibbs sampling may be used in one of two modes: i) pure sampling (constant T), or ii) optimization (T gradually decreases to help faster convergence).

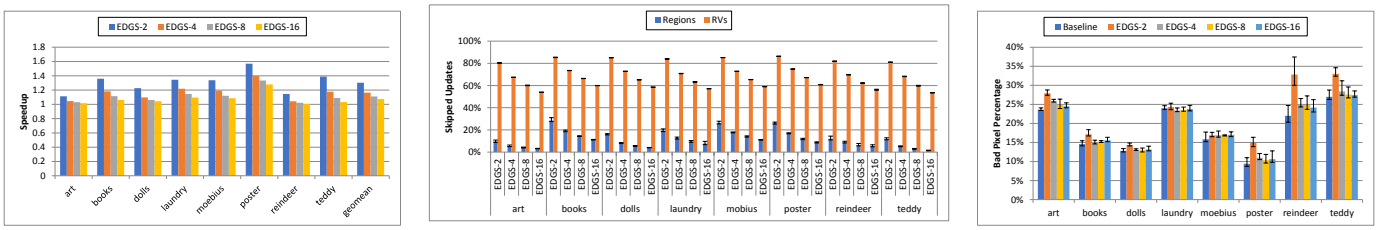
C. Approximation in Gibbs Sampling

In the optimization mode, convergence to the final result can be further accelerated by utilizing approximation techniques. One such approximation inspired by a hardware Gibbs sampling accelerator [5] is truncating very small label probabilities to zero. Equation 1 illustrates this approximation, where C is the cutoff threshold used for truncation.

$$P_{tr}(l) = \begin{cases} P(l), & \text{if } \frac{\max_{l \in L} P(l)}{P(l)} < C \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

D. Stable Random Variables

As RVs gradually converge to their final labels, it becomes unnecessary to update all RVs at every iteration because some of them simply cannot change their label. Based on our experiments, in stereo vision for the benchmark inputs only $24\% - 46\%$ of RV updates result in changing labels. Consequently, there is opportunity for up to $54\% - 76\%$ speedup. However, not all of this speedup can be gained. If a RV simply *does not* change its label does not necessarily mean that it *cannot* do so.



(a) (b) (c)

Fig. 2: Speedup of EDGS (a), percentage of skipped updates of EDGS for region- and RV-level granularity(b), and bad pixel percentage (c) for stereo vision. EDGS-C in graphs means EDGS with cut-off threshold C.

III. Event-Driven Gibbs Sampling

If a previous RV update computation resulted in a PDF concentrated on one label, which is likely due to the decreasing T in optimization mode and the probability cut-off technique, the RV's PDF is going to remain that way until something in its neighborhood changes. Therefore, we update a variable in two cases, i) if at least one of its neighbors change, or ii) it did not have a concentrated PDF to begin with. We call this optimization event-driven Gibbs sampling (EDGS). This technique is similar to vertex programming in graph algorithms, but we customize it for the context of MRF inference with Gibbs sampling. We utilize two queues to keep track of RVs that must be updated in alternate rounds (i.e., a queue for black RVs and another queue for white RVs). While updating a RV, we compare its new label with its old label, and if the two labels do not match, we put all the neighbors in their corresponding queue. Additionally, we have to check if the neighbors have concentrated PDFs. To do so, we use a matrix whose entries correspond to RVs and are set only if the corresponding RV has a concentrated PDF.

IV. EDGS Implementation for GPUs

To evaluate the effectiveness of EDGS, we implemented it for execution on GPUs due to the massive parallelism provided by them. Due to the single-instruction multiple-thread (SIMT) execution model of the GPU, i.e., all threads in a warp execute more efficiently when they are in lockstep, skipping updates of stable RVs is better done at the granularity of warps instead of individual RVs. To address this issue, we break the MRF into regions at least as large as a warp and keep track of the conditions for updating RVs at the granularity of these regions instead of individual RVs. Although tracking RVs at a coarser granularity might decrease the opportunity to skip updating stable RVs (because a region must be updated even only one RV in it is not stable), an upside of this approach is lower pressure on the memory system. Since we use queues to store the indices of RVs to be updated, tracking regions instead of individual RVs means that much smaller queues are needed. In addition to smaller queue size, there will be less contention for queue operations.

V. Evaluation

We use stereo vision with input sets from Middlebury [3] for our evaluations. Our baseline is a parallel version of Gibbs sampling with no skipped RVs. We performed design space exploration using region size, thread blocks per SM, and probability cut-off threshold. We selected the fastest point for each design in our comparisons. We ran all experiments on Nvidia RTX 2080 Ti GPU. We use bad-pixel (BP) percentage as the quality metric. We also report the percentage of skipped updates using regions of RVs to quantify the effects of EDGS on the Gibbs sampling algorithm performance.

Figure 2b compares the percentages of skipped updates for two granularities of region-level and RV-level. The figure illustrates the trend of declining skipped updates as the cut-off threshold decreases for both granularities, which has the effect of smaller speedups as the cut-off threshold shrinks. Furthermore, it shows the best case opportunity for skipping updates if we were to track update conditions at RV-level instead of region-level. Although at a finer granularity the opportunity for skipping updates would grow by 52.5%-82.2%, the mismatch between the SIMT execution model and RV-level updates leads to 10.9%-43.5% slowdown. On the other hand, the highest speedup for region-level updates is 30.3% which is achieved by EDGS-2. However, it comes with a high quality loss in some cases. EDGS-4 limits this loss to less than 3% and provides a 16.4% speedup. Evaluations with more applications are presented elsewhere [1].

Acknowledgements

This project is supported in part by Intel, the Semiconductor Research Corporation and the NSF (CNS-1616947).

References

[1] R. Bashizade, "Accelerating data parallel applications via hardware and software techniques," Ph.D. dissertation, Duke University, 2020. [Online]. Available: https://hdl.handle.net/10161/22190.

[2] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[3] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, Dec 2001, pp. 131–140.

[4] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 6, pp. 1068–1080, 2008.

[5] X. Zhang, R. Bashizade, C. LaBoda, C. Dwyer, and A. R. Lebeck, "Architecting a stochastic computing unit with molecular optical devices," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, June 2018, pp. 301–314.