Stochastic Architectures for Bayesian Inference

Eric Jonas
MIT & Navia Systems
jonas@mit.edu
jonas@naviasystems.com

Vikash Mansinghka
MIT & Navia Systems
vkm@mit.edu
vikash@naviasystems.com

Stochastic Circuits for Monte-Carlo Inference

We have developed natively stochastic circuits, which operate by producing samples from probability distributions conditioned on their inputs; this lets them obey abstraction and composition laws that closely mirror the probability algebra. This enables construction of large-scale systems for probabilistic inference. Using these rules, we show how the conditional independence structure of many probabilistic problems provides ample opportunities for parallelization.

The uncertainty present in most probabilistic problems dwarfs any rounding error or precision errors present in the computational substrate, allowing us to compute with very low bit precision. Because our stochastic systems generally evolve over time, transient perturbations rarely impact long-term behavior. We have performed quantitative experiments demonstrating very low degredation in accuracy (measured via KL divergence) at very low bit precisions (e.g. 6 bits), as well as a high degree of robustness to transient faults (in some cases including rates 1,000,000 times higher than those tolerated by conventional deterministic architectures).

We have shown how constructing stochastic finite state machines via the above stochastic elements result in architectures ideally suited for Markov-Chain Monte Carlo approximate inference techniques. We have demonstrated this via three example systems: an automatic, parallelizing compiler for compiling factor graphs down to silicon, a probabilistic video processor for low-level-vision MRFs, and a stochastic streaming architecture for nonparametric mixture models. All of these examples operate over one thousand times faster than what is possible with existing processors. We hope that these approaches will encourage the application of fully-Bayesian reasoning to low-power, embedded, and real-time applications where they were previously believed unfeasible.

Circuit Compilation

Because of our abstraction and composition laws, automatic transformation (compilation) from a high-level language to a synthesizable circuit becomes possible. We have developed a compiler for transforming arbitrary-topology discrete-state factor graphs into synthesizable densely-parallel circuits capable of performing inference at millions of samples per second. The compiler automatically identifies the conditional independence structure in the model to exploit opportunities for parallelism. Example problems have been compiled and synthesized, including a classic Bayesian Network and an Ising model from statistical mechanics.

Lattice MRFs and real-time optical flow

We have built a real-time optical flow engine using a probabilistic video processor that enables dense optical flow computations from live video, by dynamically swapping in and out parts of a lattice factor graph. The engine we describe can produce 100000 samples per second from the joint distribution of latent variables. It is weakly-reconfigurable, in that it can be adapted for almost any lattice MRF.

Nonparametric Architectures and Clustering

By using stochastic queues and storing drastically-reduced quantities of data (the "sufficient statistics"), it's possible to construct highly-efficient stochastic circuit for nonparametric mixture models. The structure of the probability model still allows us to exploit fine-grained parallelism, resulting in an architecture that is 1000 times faster than a conventional processor at a tenth of the power. This enables us to cluster MNIST digits in real-time, with classification performance competitive with discriminative methods. This architecture scales to clustering millions of data points with thousands of features.