

Title: Probabilistic Programming

Link: <http://research.microsoft.com/pubs/208585/fose-icse2014.pdf>

What are the problems/research questions addressed by this article?

--->

The author describe connections this research area called "Probabilistic Programming" has with programming languages and software engineering, and this includes language design, and the static and dynamic analysis of programs. We survey current state of the art and speculate on promising directions for future research.

What are the existing solutions for this research question/problem?

--->

The idea has been around since the 80's in the past decade there has been a lot of progress in building model checkers such as PRISM [36] and MRMC. PRISM supports various kinds of probabilistic models such as Discrete Time Markov Chains, Continuous Time Markov Chains and Markov Decision Processes.

What is the research method [s] they have used?

--->

The author start by discussing the relationship between Probabilistic Programming and Probabilistic Model Checking. They explore scope for interaction with other sub-communities in machine learning such as the optimization community .

What is their proposed solution ?

--->

The goal of a probabilistic program is to represent (and model) a probability distribution. The view point taken is Bayesian, and typically a probabilistic program assigns to variables from a "prior" distribution, and then constrains the relationships between variables using observe statements, and the goal of the program is to represent a "posterior" distribution obtained by conditioning the prior distribution using observations. The goal of a probabilistic inference (as implemented in a probabilistic programming system) is to compute appropriate representations of this posterior distributions, or expectations of functions with respect to this posterior distribution. In contrast, the goal of probabilistic model

checking is to perform verification, that is, to model a system with probabilistic components and verify properties that place quantitative bounds on all possible behaviors of the system.

What are three future directions from this article?

-->

While the goals of the communities have been different, we believe that cross-fertilization between the two communities is an interesting direction for future work. Compilers need to implement the equivalent of common compiler optimizations for probabilistic programs, and runtimes need to exploit the power of massive parallelism available in today's GPUs and cloud services. Diagnostic information needs to be provided to the programmer to help her identify programming errors, and improve the quality of the programs she writes. Substantial improvements in these areas will need interplay between compiler analysis, machine learning and usability research with programmers and data scientists.

Concepts that you learnt from this paper?

--->

Nondeterminism.

Dynamic Analysis.

Loopy Probabilistic Programs.