Extending Resource-Aware ML with Probabilistic Programming

Evan Wu

December 18, 2019

Project website: https://www.andrew.cmu.edu/user/ejwu/

1 Project Description

Who: I will be working with Jan Hoffman, a professor, and Di Wang, a graduate student.

What: Resource-Aware ML is a tool that automatically determines cost bounds for OCaml programs. This research will aim to extend RaML to determine expected cost bounds for probabilistic programs. Naturally, probabilistic primitives will have to be implemented and a syntax and semantics constructed for them. What will be challenging about this project: The most time consuming part of this project may be the large code base of RaML. It will take some time to familiarize myself with how it works, as well as implementation details. In addition, we need to integrate the probabilistic features with RaML, which may present problems even if the semantics on paper is fine. There is already a clear strategy for how to construct bounds (see papers in Literature Search section). The most difficult part will be the implementation.

So What: If we complete this project, we will prove that it is possible to automate the analysis of probabilistic programs. The methods used are interesting theoretically, and the method could be extended to other languages and inspire future research. With further development, this research could help library authors more quickly verify resource bounds without tedious manual checking.
2 Project Goals

100% goal: I plan to add syntax and semantics for probabilistic programming to RaML, and implement them. I will produce a tool that automatically derives resource bounds for probabilistic programs. I plan to prove the correctness of the bounds obtained by our analysis. I will analyze a reasonable range of different programs and algorithms to demonstrate the effectiveness of our tool.

75% goal: I will have a proof, but no implementation of our tool. I will have example programs or algorithms but only analyze them on paper.

125% goal: I will incorporate more features from RaML and its extensions, or add more features to the type system. In general, we would increase the expressivity of our syntax and semantics so we can analyze a wider range of programs.

What metrics can we use to determine the success of the project?
We would really like to define a syntax and semantics, and be able to algorithmically derive resource bounds, even if it is just on paper. Beyond that, we would like to be able to analyze as wide a range of programs as possible. Also, as discussed above, we would like the actual implementation to be as complete as possible.

3 Milestones

1. Read about and understand RaML. I will read the lecture notes Jan provided for this purpose.

2. Read about probabilistic programming. Specifically, read lecture notes on Jan’s website and the two papers in Literature Search and understand the topic.

3. Finish soundness proof started by Di. This is a proof that the bounds given by our tool are consistent.

4. Apply our work to example algorithms on paper. Investigate limitations of our work. Investigate what cost bounds we obtain. Think about possible improvements. [expand]

5. Familiarize myself with the RaML code base. Read through and understand the overall structure of the code well, so that I know how to add to it later.

6. Add syntax and dynamic semantics for probabilistic programming to RaML. We should be able to run probabilistic programs written using our new syntax.
7. Implement static semantics and analysis. We should have a type-checker and be able to derive resource bounds on a reasonable range of programs.

8. Project Report/Draft. This milestone would mostly consist of a very rough draft. There is the possibility of a paper, in which case we will want to take time to draft the paper. If there is no paper, we will still want to write up our results and methodology, in preparation for possible future research or a future paper.

4 Literature Search

The following papers deal with cost analysis of probabilistic programs using a similar strategy to this research, that is, constructing inequalities for worst-case bounds from the program and feeding them into a Linear Programming solver.

- Cost Analysis of Nondeterministic Probabilistic Programs [1]
- Bounded Expectations: Resource Analysis for Probabilistic Programs [2]

To familiarize myself with RaML, as well as have a detailed explanation of how cost analysis of probabilistic programs can be implemented, we have Jan’s lecture notes from 15–819: the relevant lectures are Weeks 6–9. I will also need to read Di’s incomplete proof, which lays out a syntax and possible semantics. I believe this will be enough to fully understand what I need to for this research project.

5 Resources Needed

Only a laptop is necessary for the project. RaML is available on the web. For running RaML, any system able to run OCaml is enough. We will also have a regular meeting to talk about progress and discuss ideas on the theory side.
References
