Term Project
CS7470 Fall 2023

Steven Holtzen
s.holtzen@northeastern.edu

1 Overview

- The goal of this project is to deeply explore one of the core areas of probabilistic programming. The project will be self-determined: you are encouraged to choose a project that you find personally interesting or inspiring that is relevant to the course.
- The project will proceed in 4 phases: (1) a *proposal* where you scope and explain the goals for your project; (2) a *check-in* where you give a brief progress update; (3) a *final presentation* where you share your work with the class; (4) a *final report* where you summarize your results. Each of these phases is summarized in this document.

2 Administrivia

Proposal Due Date Friday, Oct. 30 at 11:59PM EST.

Check-in Due Date Friday, Nov. 20 at 11:59PM EST.

Final Due Date Friday, Dec. 11 at 11:59PM EST.

Late Policy Late work with no prior permission from the instructor will not be accepted. Please contact the instructor prior to the deadline for an extension, which can be provided on a case-by-case basis.

Group Work You are permitted, but not required, to work in groups of up to two people. A group will turn in a single term project. You may consult with others throughout the course of your term project; if you do so, please include their names in your final report.

Academic Honesty You are encouraged to use open-source code and libraries as long as you respect the licenses of those libraries. In particular, be sure to attribute any code that you use from an open-source library appropriately: *do not copy and paste someone else's code without properly citing its original source*. When in doubt, ask the instructor or consult the Northeastern academic integrity policy.[1]

2.1 Rubric

The project will be evaluated on the following rubric:

- 10% Project Proposal
- 10% Project Check-In
- 20% Final Presentation
- 60% Final Writeup

[1] http://www.northeastern.edu/osccr/academic-integrity-policy/

The proposal, check-in, and final presentation steps will be evaluated on a pass/no-pass basis. Details for the final write-up rubric are given in Section 5.

3 Proposal

The proposal is a single document that should contain the following information:

1. Include your name, a project title, and any group members.
2. Summarize your chosen project idea. Why have you chosen it?
3. Briefly justify its relevance to the course.
4. What will be the outcome of the project (a code artifact, a set of attempted benchmarks, etc.)?
5. *Measures of success*: What are your measures of success for your project? Provide the following:
   - Baseline: What is the minimum level of progress at the end of the quarter to be considered a success?
   - Medium: What would be considered an unqualified success?
   - Stretch: If you have extra time, what might you consider working on?

   These are to help you scope the project, and are subject to change during the quarter.
6. What challenges can you foresee in completing your project? What resources might you need in order to be successful?

If you need to change your proposal in a major way after you submit it, please contact the instructor.

3.1 How to Choose a Project

- Choose milestones carefully: try to choose a project with a balance between attainable goals and interesting reach-goals.
- An interesting failure is better than a predictable success. Even if things really go badly and don't turn out as you were hoping, in the end a valuable project contribution is to be critical and explain precisely how things went wrong. This can be an important learning experience as well.
- When in doubt, ask for help: Steven is happy to brainstorm ideas during student office hours or over email.
- You are not strictly required to have a programming component for your final project, but it is highly encouraged.

Some project ideas Here are some ideas for projects. You are encouraged to come up with your own project ideas; when in doubt, contact the instructor, who is happy to help you brainstorm.

- Extending our mini discrete or continuous language with new features or inference algorithms (i.e., differentiability, loops, functins, etc.).
- Implementing or extending an idea or method from one of the papers that we read.
- An application of probabilistic programming to your area of research or interest.
- Adding a new feature or extension to an existing probabilistic programming language.
- Implementing a static analysis or interesting type system for one of our small languages.

4 Check-In

The check-in is a single document that should contain the following information:

1. Include your name, a project title, and any group members.
2. Give a brief summary of your progress so far.
3. Have you needed to change or adjust your goals? How?

5 Final Report

The final report is a more open-ended document than the previous two documents. This document as a hard limit of 6 pages not including references, with unlimited appendices. It should follow a standard research paper structure. The following rough structure is suggested but you may use whatever structure you like:

- *Abstract*: Describe the contents of the project in about 200 words.
- *Introduction and Motivation*: Explain the selected problem and describe why it is interesting and related to the course. Outline technical contributions.
- *Background*: Describe the relevant background material for understanding the technical work.
- *Technical Contribution*: Explain what you did for your project in detail. Describe your experiments or theoretical work.
- *Related Work*: Describe related work.
- *Conclusion*: Describe future work. Did you succeed in your initial objectives? What would you do differently next time?
- *Feedback*: Please give the instructor some feedback on the project. What went well? What would you prefer be done differently next time?

Your final writeup will be evaluated on the following metrics:

- *Quality of Writing*: Are the ideas clearly presented and easy to follow? Is there appropriate usage of figures, tables, and other non-textual elements?
- *Technical Quality*: Is the work technically sound? Is there a scientific hypothesis? If there are experiments, are they well-designed to answer scientific questions? If there are theorems and definitions, are they clearly stated and proven correct? If code is provided, does it work? How does the final product compare to what was discussed at the checkpoint?
- *Scholarship*: Is the appropriate related work discussed and compared against? Is the work well contextualized in the field?

6 Final Presentation

Each group will give a final presentation to the class about their project:

- You are encouraged to use slides to give you presentation.
- Walk through the elements of your project given in the final report.