Specification for CMPUT 609 Course Project Reinforcement Learning II

February 8, 2020

Your class project should be an empirical investigation of an issue explored in the course, or the combination or intersection of two issues explored individually in the course. In particular, the issue should arise in the textbook in Chapters 9, 10, 11, 12, 13, possibly 14, or in the earlier sections we covered, Sections 5.8, 5.9, 7.4, 7.5, or 7.6. The project should not involve any completely new issues not covered in the course. There are three primary objectives: 1) to pose a scientific hypothesis that builds on but is distinct from the material in the textbook, thus testing your ability to see the edge of what is already known in the field; 2) to train you in designing, conducting, and concisely describing empirical experiments that add to existing knowledge; and 3) to implement a complete reinforcement learning agent and task. These three are the most important component skills to conducting research in reinforcement learning.

You should structure the project as an empirical reinforcement learning experiment or sequence of experiments. To begin, you should have a hypothesis that you can state clearly and that is interesting to some extent. A classic hypothesis is to ask which of two algorithms, or two variations on an algorithm, is better on a particular reinforcement learning problem. To properly answer such a question you will need to try each algorithm with a range of values of its parameters and assess the statistical significance of any differences you find.

Many of the experiments described in the book are good examples or foundations for projects. They are mostly simple, clear, and unambitious, yet they are instructive; try to make you project like these. In fact, a good project could be made based on most of the examples in the book. First replicate the given results, then extend them in some instructive way,

perhaps by using more algorithms, or a variation on the task, or both. One caveat: The experiments in the book are often carefully chosen so that statistics are not necessary to interpret the results, whereas usually some statistics are necessary in a scientific project; I want you to get some experience with this.

A good project need not be unique to an individual. It should address something of recognizable, arguable interest to many, to a generic reader. It should be of interest such that we care about the results, that we want to get them right and we are genuinely unsure in advance of how they will turn out. It is fine, perhaps even desirable, for multiple students to work on the same issue, and for them to talk to each other about it. But each student must perform their own experiments and write up their results independently.

1 Good and Bad Projects

Your project should not involve a new domain that does not fit neatly into the MDP framework. Your project should not involve a new domain that involves extensive explanation.

Your project should not involve multiple novelties—just one new aspect please. In any event start small and see where one novelty takes you. Often such things expand and ramify. You can always add an additional novelty later if you fully finish with the first novelty.

The real rule is that you want your project to have some meaning, to answer some question and not just bring up new ones without significant progress having been made. You want to close off some question or questions.

A good project says, here is an interesting issue that has come up, and here is a straightforward series of experiments that could be used to possibly resolve it. Here, I've done those experiments; here is what happened; this is what I conclude from them with regard to the interesting issue. In doing such a project you should learn (or at least get some experience with) how to frame a scientific question, how to reduce it to an experiment, and how to assess your results, extracting the maximum of meaning from them without going beyond what has actually been shown.

A bad project says, what about this? This was not covered in the book! I will make a new project about it. But then the project will not build on what you have learned in the course in any way. I don't want to see this.

A bad project says, here is a new domain or problem. What if you wanted to solve it in this sort of way? How could that be fit into the RL/MDP framework, even if it does not look very suitable? Then the project will be

all about the domain and again not really build on what you learned in the course. A new problem can be good *as part* of a project *if* the problem fits well into the the RL/MDP framework and is well suited to the issue being explored.

2 Risks

The most common mistake is to try something too ambitious. Experiments with reinforcement learning agents are actually a bit more difficult than they probably seem at this point. There are parameters, and there are many issues in the setup of a task, in the choice of reward, and there are many many issues in representations and choice of state space. The importance of all these things is understated in the book and is thus probably not apparent. Perhaps because of this, a common mistake is to start a project that is too ambitious, that addresses some large problem involving many AI issues. In this case, the project is unlikely to have a successful outcome in the available time.

To deal with the risk of this mistake, I urge you to design a simple project, one that can be done quickly in an initial form (and perhaps then extended), and that goes just a little beyond what is currently known. Another nice thing about a very straightforward project is it would be easy for me to understand your results.

The second most common mistake is to not leave enough time to get good experimental results. Doing a good experiment and getting meaningful results is not easy. It is harder than you think. It always seems to be the case that when the time comes to submit a paper or a project, that I would like to run the experiments all over again one more time, maybe to make the results more statistically significant, or maybe to get rid of an annoying detail that I would rather have done a little bit differently. Or to run one more comparison to better understand what happened. I put a big emphasis on clarity of results. It is much better to have a small clear result on a small problem than a vaguer result on a large problem where it is not quite clear what it means or why it happened.

In the past, when I've asked a class to do a project involving a little bit of research, the most common mistake has been not to have a clear hypothesis—not to have a clear question being asked and tested by the experimental research. Formulation of hypotheses is something that we should understand completely if we see ourselves as computer scientists. Your experiment should have a clear scientific hypothesis. Your experiment

should clearly test or bear on that hypothesis. But the first step (hypothesis formulation) is often difficult for students. Formulating the hypothesis is a difficult thing. Not difficult because it's hard like lifting a heavy rock, but hard because you have to think clearly, and hard because you have to give up on all the possible things you might do and pick one question to clearly ask. It's much easier to ask or wonder, "what would happen if I applied reinforcement learning to some large problem?" Or ask "what if I try to make this idea work?" Or "I wonder what would happen if I tried so and so?" These are not good scientific hypotheses. Instead, you have to ask something small and precise. Like, "in problem A, is Sarsa better than Q-learning" or "does the best step size scale linearly or quadratically with the number of features in a state representation?" Or "in off-policy learning on such-a-such problem, do we see instances where TD(λ) diverges, but the new gradient TD algorithms converge?" Or "on this problem, which of these two kinds of prioritized sweeping are more effective?"

3 The Writeup

An important part of your project will be to produce a complete writeup of the experiment(s). This is not an easy thing to do well, and I want you to do it well. The writeup need not be long. You should write just enough to mention the extant issues and what new is being done with them. Lean on the book to reduce the writing. You don't need to repeat what is there; instead just incorporate it by reference. But be sure to adhere to all of the following requirements, on which you will be marked:

1. You must fully explain the experiment(s) such that I could give your report to one of your classmates and they could reproduce the whole experiment(s) and get exactly the same results (excepting possibly that they would have a different sequence of random numbers).

2. You should clearly separate the presentation of the problem from the presentation of the learning algorithms that you apply to the problem.

3. You should clearly separate your presentation of the results on the problem from the conclusions that you draw from those results. The best way to do this is by switching tense.

4. You should use present and past tense properly. Algorithms and problems both just *are*, and they should both be described in *present tense*. Your experiment, on the other hand, is something that you *did*. It and

the results you obtained should be described in the *past tense*. Finally, any conclusions that you draw from the results should be described in present tense again. For example, an algorithm maintains (present tense) an approximation to the action-value function, and a problem has (present tense) a 3-dimensional state space with 4 discrete actions, and the actions affect (present tense) the state is such and such a way. In the experiment, you applied (past tense) 10 instances of the algorithm, each with a different value of the step-size parameter, to the problem. Each algorithm instance was initialized (past tense) with a particular weight vector, and then run (past tense) for 100 episodes. Then the whole thing was repeated (past tense) for 30 runs. The random seed was initialized (past tense) to the same value for all algorithm instances at the beginning of each group of 30 runs. For each run, the total reward on each episode was recorded (past tense) and averaged (past tense) over runs to produce the learning curves shown in Figure 1. I conclude that the differences between algorithm A and algorithm B at episode 100 are (present tense) statistically significant because they are more than twice the standard error in both means. Thus, on this problem, algorithm A clearly performs (present tense) better than algorithm B, though it remains unclear whether it performs (present tense) better in general (i.e., on other problems). Further experiments on other problems would be needed to make that conclusion.

5. You will probably need to vary parameters to see how well each algorithm can do. To show that one algorithm is better than another, you normally would have to show that it is not difficult to find parameter values for the winning algorithm that enable it to perform better than the losing algorithm does at any setting of its parameter values.

6. In presenting your results, you will probably want to have a sequence of graphs. Typically, early graphs present more detail and validate the basic idea of the experiment, and then you progress to more summary and all-encompassing graphs, just as in the labs we did learning curves then plotted summary numbers for each learning curve as a function of the step-size parameter. In general, it is helpful to the reader to present your results incrementally, gradually adding complexity and summarizations.

4 Schedule

now	Start thinking about possible project ideas; discuss them with your classmates; write in your research diaries about them; try to formulate a clear hypothesis.
Reading week	Conduct the experiments to test your hypothesis. Revise and clarify your hypothesis as needed. Repeat to maximize the cleanliness and elegance of your experimental design.
Feb 24 – Mar 4	Write and rewrite your project description.
Thursday, March 5	Project due and submitted via eclass.