CS6700 : Reinforcement Learning
Programming Assignment #1

Multi-armed Bandits — Deadline: 02^{rd} Mar 2019, 11:55 pm

- This is an individual assignment. Collaborations are strictly prohibited.
- You have to turn in the well-documented code along with a detailed report of the results of the experiments. Typeset your report in LaTeX.
- Be precise with your explanations. Unnecessary verbosity will be penalized.
- Check the Moodle discussion forums regularly for updates regarding the assignment.
- *Any kind* of plagiarism will be dealt with extremely seriously. Acknowledge any and every resource used.
- **Please start early.**

You are to conduct experiments on the 10-arm bandit testbed described in Section 2.3 of the book. Please turn in the code for the testbed as well as the algorithms. Label the graphs clearly, with axes labels, parameter values, question numbers etc. Ensure that the code is adequately commented. Also turn in a short write-up of your observations from the experiments and answers to all the questions asked below.

Remember: The graphs are to run for 1000 plays, with each curve being the average of the performance on 2000 different bandit problems, generated as per the description in the book.

Points will be given according to the following criteria :

- Correct generation of the bandit problems.
- Correct coding of the learning algorithms.
- Correct code for gathering data to plot the graphs,
- Performance of the learning algorithms (correctness, optimality),
- Neatness of the graphs, appropriate labelling etc, and well-commented code.

Note: You can program in any language you want, but if you need any help later on, it will have to be in a language the TAs are comfortable with. So check with us before you start.

1. (10 marks) First implement the 10-arm bandit testbed. Try to reproduce the graphs given in Figure 2.2 on page 23 of the textbook. Specifically, implement the ϵ-greedy method on the 10-armed bandit testbed, for different values of ϵ.

   Use graphing software of your choice, (for eg. Matlab or Gnuplot), to produce the graphs. Ensure that you have labelled the graphs correctly.

2. (8 marks) How does the softmax action selection method using the Gibbs distribution fare on the 10-armed testbed? Implement the method and run it at several (at least 3) temperatures to produce graphs similar to the previous one. Note that now you are required to sample from the softmax distribution, and not take the action with the maximum probability!

3. (7 marks) Implement the UCB1 algorithm. Compare the performance with that of epsilon greedy and softmax. What do you observe? Why do you think this is so?

4. (5 marks) What happens as the number of arms grows? Run experiments on a 1000 arm bandit setup and compare.

You are hired to work for an advertisement agency of a large organization. Every new visitor to your organization's website is played one of three advertisements. For every advertisement you suggest, the suggestion team will provide you with a score, rating the appropriateness of the delivered advertisement.

You are required to suggest the most appropriate advertisement for each users(the most appropriate ad will receive the highest score from the suggestion team). The users are presented as a sequential stream and every user is represented as a 4-dimensional vector.

---

5. (10 marks) Formulate this as a contextual 3-arm bandit problem and use Reinforce to quickly converge upon a solution from samples.

A template python program is present in the zip file *problem1_5.zip* uploaded to moodle. You do not have to modify ADS.PY and ADS.CSV, but only TEMPLATE.PY. Generate a plot of the training curve (reward vs. time step).

Submission Guidelines

Submit a single **zip** file containing the following files in the specified directory structure. Use the following naming convention: `rollno_PA1.zip`
A sample submission would look like this:

```
rollno_PA1
├── Code
│   ├── experiment_testbed_file(s)
│   ├── algorithm_file(s)
│   ├── plotting_file(s)
│   └── ...
├── report.pdf
└── README
```