

## Programming Exercise 3

Consider a puddle world exactly like in the previous assignment, but blown up by 1000 times. While it is still possible to run your earlier code on this version, the aim of this exercise is to get you to play around with function approximation.

1. Repeat problem 1 from Programming Assignment 2, with a linear function approximator. You can try a simple CMAC if that is convenient. [Bonus: How do you think the parameterization given in question 3 in previous assignment for the policy will fare in case of value function approximation?
2. Repeat problem 2 from Programming Assignment 2 with a linear function approximator. Report results for at least 2 values of  $\lambda$ .
3. Compare the performance of Sarsa( $\lambda$ ) with and without function approximation on this task. Specifically note the rates of convergence and the quality of the final policy.

## Evaluation Criteria

The points will be given according to the following criteria:

- Correct coding of the puddle world dynamics
- Correct coding of the learning algorithms
- Compliance with RL-Glue
- Correct performance of the learning algorithms
- Neatness of the graphs - correctly labeled etc. and well commented code
- Correct derivation of policy gradient updates and thoughtfulness of comments

**Note:** You can program in any language you desire, but the RL-Glue benchmarking routines can work only with in C/C++, Java, and Python (and other C-callable languages). For other languages, you might need to recode the benchmarking routines also.

You have to turn in a lot of learning curves. Organize them in a sensible fashion, across all parts, so that the number of graphs you turn in is minimal, yet each graph remains comprehensible. :-).

The due date for this assignment is **Tuesday, March 28th**.