

CS 4100/5100: Semester Project

Due: December 14, 2015

1 Project Description

The final project should be an application of some area of AI to a problem of interest to you. For example, you might develop a game playing agent for a simple version of poker, blackjack, or some other card game. The game playing agent might be based on adversarial search or it might use reinforcement learning. Alternatively, you might use constraint satisfaction methods to develop various puzzle-solving agents. In the past, students have sometimes explored applications of classification to machine vision problems. It doesn't matter so much what application you choose to study. The key requirement is that you think about how the methods we studied in class (or perhaps other methods that we did not study!) can be applied to a real-life problem.

You may work alone or in groups of two or three.

2 Timeline and Deliverables

11/5/2015 Project proposal due. Please submit a one-paragraph document describing a proposed problem and solution via SVN. There should be a `./project/` directory in your SVN account. We will review all project proposals. Some projects will be given the go-ahead via email while we will ask to meet in person with other project groups.

12/4/2015 Final project due. You must submit a three or four page project report (preferably in PDF format). Also submit the code used to produce your results as a separate directory in the project SVN directory. Your report should include the following:

1. **Problem description:** What problem are you solving? Describe the problem from a computational perspective. What are the inputs and outputs (exactly)?
2. **Algorithms:** What algorithms do you use? Why are these algorithms appropriate? How are these algorithms typically used, and how are you using them?
3. **Results:** Quantitatively characterize how your algorithm worked. Under what circumstances does the algorithm solve your problem successfully? When does it fail? You should quantify performance somehow. For example, if you are using a search algorithm, you might report the number of nodes expanded for different problem scenarios. If you are using a learning algorithm, you might report accuracy. If you are using reinforcement learning, you might report average utility as a function of trial number.