

15-381: Artificial Intelligence

Assignment 6: Mini-Project

Fall 2001

Proposal due: Monday, November 18, noon (WeH 7123);
Project due: Tuesday, December 11th (in class)

This last homework is a mini-project exploring a topic of your choosing. The mini-project is to be a more substantive bit of work than a standard homework, so it is a good idea to get started early. A mini-project is an opportunity to further explore a topic which you found interesting. Mini-projects typically involve coding up some algorithm covered in class and evaluating its performance on some example problem or data set. You can either select a project from the list which follows, or propose your own.

Mini-projects can be done either individually or in groups of two. (If you want to have a group of three people, please talk with the instructors.) We highly recommend working in teams. All team members will receive the same grade for the project—it is up to you to divide up the work.

As a first step, you need to write a mini-project proposal. The proposal should be roughly ONE PAGE and include the following:

- A list of group members.
- A description of the topic you plan to explore.
- A description of the problem you you plan to solve.
- A description of the methods your solution will involve.

Once you have completed your mini-project, you need to hand in a project report. The report should contain the following:

- A description of the algorithms/methods implemented.
- A description and analysis of the results obtained.
- A description of the lessons learned and your conclusions.

In addition, you need to hand in the code that you wrote for the mini-project.

1 Candidate Project Topics

The following is a list of sample project topics. You may choose from one of the topics in this list, come up with a variant, or come up with a completely new idea of your own.

1.1 Search

Apply A* with different heuristics to a difficult problem, such as route planning.

1.2 Comparative analysis of two or three text classifiers

Go to a web site such as Yahoo that has web pages categorized into taxonomy. Select 3 to 4 categories in the taxonomy, and for each category ensure that you have at least 30 test web pages with text on them (e.g. *not* mostly GIF or Javascript). Divide the collection randomly into a training set, a validation set, and a test set. Then implement two or three text categorizers (e.g. each group member implements one), train them on the training set (with category labels) and test them on the validation and test sets (without labels) to see how well they predict the category label. Use accuracy, precision, recall, and/or F1 (from the class notes) as your measures. Which classifier performs best? Why might this be so? Try different random splits of the data and analyze which is more sensitive to the selection of training/testing data? [You may use Naive Bayes, k -Nearest Neighbor, Decision Trees, or any other such classifiers.]

1.3 Document Clustering

Select web pages as in the previous topic, but instead of labeling the pages apply document clustering techniques to group them. You should implement the k -means clustering algorithm and experiment with different distance measures (e.g. cosine similarity, Euclidean distance, kernel functions). How well do the clusters correspond to the original categories?

1.4 Implement alpha-beta pruning to play Connect-4

Implement mini-max with alpha-beta pruning, and apply it to the game connect-4. You will need to come up with a proper evaluation function. Alternatively, you could use machine learning techniques to learn a good evaluation function. The end product should be an implementation of the game with enough of a user interface (can be text based) so that a person can play a game against a computer opponent, or view the computer play against itself.

Connect-4 is a simple two-player game in which each player has a stack of colored chips—one color per player. A 7×6 board is placed between the two players, propped up vertically. It has seven “slots” at the top—one slot for each column. When a game chip is dropped into one of these slots, the chip slides down until it either rests at the bottom of the board or on top of another chip. The two players take turns dropping a chip into a slot of their choice. The first player to connect four of his/her color in a row (either vertically, horizontally, or diagonally) is the winner.

1.5 Game playing

Apply mini-max with alpha-beta pruning to another 2-player game that may interest you. (Note that chess, bridge, or Go may be too hard for this short-term project, but you may try!).

1.6 Planning

Solve a puzzle, such as RushHour or Lunar Lockout, by develop your own planning algorithm or use one available planner (GPS, Prodigy, SNLP, GraphPlan, SATPlan, or FF - we will be able to provide pointers to code for these planners.

1.7 Supervised Learning - Classification

Select any data set with labelled data and apply a supervised learning algorithm to it - decision tree learning or neural network learning. Examples of databases of feature-valued data are in the Machine Learning repository of the University of Irvine.

1.8 Machine vision and machine learning

Apply at least two machine learning techniques to a machine vision task like face detection, car detection, or head pose detection. Analyze and compare the performance/features utilized by the machine learning algorithms. Some possible choices of machine learning algorithms which could be applied here are neural networks and Naive Bayes.

1.9 Frame/Semantic-net interpreter

Implement a frame/semantic-net interpreter with pushers and pullers as well as simple inheritance.

1.10 Apply simulated annealing to TSP

The Traveling Salesman Problem—TSP for short—can be stated as follows:

A salesman spends his time visiting n cities (or nodes) cyclically. In one tour he visits each city just once, and finishes up where he started. In what order should he visit them to minimize the distance traveled?

Apply simulated annealing to find (suboptimal) solutions to TSP with 100-200 cities randomly distributed in a Euclidean space. Assume that the distance between two cities is just the Euclidean distance between the coordinates of the cities.

1.11 Implement DPLL with heuristics

Implement the DPLL algorithm for solving constraint satisfaction problems. Try different variable ordering and value ordering heuristics, and report which seem to work best. Suggest other search heuristics that could be useful.

1.12 Q-learning

Implement the Q-learning algorithm and apply it to world models with non-deterministic effects, and with different reward structures. Vary the discount factor, and try different experimentation strategies (exploitation vs. exploration). We will be able to provide a simulated world.

1.13 Choose your own topic

Be creative and define your own project. Select some cool AI-related topic that you found really interesting during the semester and would like to learn more about. Think about what you want to do (it needs to be a mini-project sized chunk of work) and describe the topic and work you would like to do in your one page proposal. You should discuss your idea with one of the professors or TAs before writing the proposal.

2 Completion of the Project

You will need to turn in a working program and a write up of no more than 5 pages. The write-up will have to be organized with the following, and only the following, sections:

- 1. Statement of the Problem
- 2. Approach (to include a high-level description of the algorithm)
- 3. Empirical Results
- 4. Discussion and Conclusion.

Please take the requirements above into account to scope your project into a feasible project. The best projects will be the ones that try to address a simple problem and go in depth in analyzing the strengths and possible limitations of the approach.