4511W, Fall-2021
Project description

Requirements

1. Projects can be done alone or in groups of two (ask me if you want to have more and the project in question).

2. Your projects need to include:
   - a programming component (either your own code or some publicly available code to solve a problem. Your choice for the language),
   - a short literature search component (i.e. read a few peer-reviewed technical papers related to the topic you are covering),
   - an experiment run by you (on either publicly available data or your own problem then analyze the results),
   - a written report.

3. A short proposal (two to three pages) is due **Nov. 23** as the 4th writing assignment, this will include:
   - team members,
   - a brief description of the problem you focus on,
   - the approach you are going to use to solve it,
   - the software you will create or modify,
   - how you intend to run your experiments
   - an estimated timeline for the work schedule
   - bibliography references to 2 or 3 technical papers

4. The final project is due **Dec. 20**. The report should be around 10 pages, (15 pages for group of 2) written in latex. It should include:
   - Title, authors, date, short abstract (i.e. brief description of main ideas).
   - (1-2 pages) An introduction describing your problem abstractly and why it is interesting.
   - (2-3 pages) A brief review of related work about this problem, discussing existing solutions in the literature.
   - (1-2 pages) A concise description of your approach to solve the problem, including algorithms details.
   - (2-3 pages) Description of the experiment design and results.
   - (1-2 pages) Analysis of the results.
   - (0.5-1 page) Conclusion/summary and future work.
   - Bibliography.
   - For group projects, indicate the contributions of the team members.

5. The project is supposed to be approximately 50 hours, including the initial research, programming, evaluation and writing. This can mean either a fairly simple topic very well analyzed or a hard project covered in much less depth.

6. You do not need to include the source code of the program used for your results. The evaluation will be based solely on the written report, but if you use publicly available code, describe and cite the source of this.

Grading

The project is worth 15% of the total class. Points will be distributed as such:

- 25% Writing
  - 10% quality of writing
  - 10% organization and structure
  - 5% Latex
- 15% Literature search - coverage and analysis
- 30% Problem and solution
  - 15% problem and approach/solution well documented and concise
  - 15% complexity of problem/solution compared to implemented experiment
- 30% Experimental results
  - 15% design and extensiveness of experiments
  - 15% analysis of results

Example ideas

Here are some possible ideas for projects. These are only suggestions, but you may do any project of your choice if sufficiently complex.

- Write an automated player for a non-trivial two-player game, like Othello or 3D Tic-Tac-Toe.
- Write a a simple spam filter based on naive Bayes probability, following the steps outlined in A plan for Spam by Paul Graham. The article describes the problem and provides an outline of the method to be used. The article provides some Lisp code, and Paul Graham provides more useful Lisp code. You do not need to do this in Lisp, any language will work, but it is particularly simple to write it in Lisp. If interested in using lisp for this, look here for more details.
- Use A* or some other algorithm to find shortest paths in a map of Minneapolis, shown here. The map data are in the file map.lisp .
  The format of the file is as follows: each line contains information about a segment of a road. Each line includes 5 numbers in parentheses. The first number is `1` to indicate a one way road or `2` to indicate a two-way road. The next two integer numbers are the `x` and `y` coordinates of the start of the segment, the next two numbers are the `x` and `y` coordinates of the end of the segment, For instance, `(1 1121 7568 1042 7545)` indicates a one way road starting at point `[1121, 7568]` and ending at `[1042, 7545]`. There are `1357` lines in the file, each corresponding to a road segment.
  You do not need to use Lisp, the lisp format is used for the data because of convenience.
- Solve the Traveling Salesperson Problem using real data. Large data sets are available at http://www.tsp.gatech.edu/index.html
- read the paper How MapQuest Works [no longer available]. Look instead at www.siam.org/meetings/alenex04/abstacts/rgutman1.pdf for an interesting routing algorithm.
- Prof. Sven Koenig has pages with descriptions of projects and related references and sometimes relevant code. For instance, look at Any-Angle Path Planning.
- The RoboRescue simulation league addresses problems related to automating disaster management. There are two subunits, the Agent Simulation and the Virtual Robots. The agent simulation focuses on large scale (city level) planning for disaster management, while the virtual robots provides a realistic simulation environment for robots. Look at https://sourceforge.net/projects/roborescue/ for information and for the simulator.
- Look at this thesis and see if anything interesting: http://emotion.inrialpes.fr/people/synnaeve/phdthesis/phdthesis.html

Possible help for writing

Here are some ways to help with the literature search:

- Go to scholar.google.com and search manually
- Go to the ACM Digital Library, which is accessible to students and faculty. If the link does not work, you can get to it from the UofM library page ACM guide to computing literature
- The AI Topics library from AAAI offers a collection of pages with information about AI and various subjects within AI. The library is organized by topics and is a good place to get a start on the subfields of AI.
- use Google with a few key terms. In general you will find lots of references.
- Go to the ResearchIndex at citeseer.ist.psu.edu. This is a reasonable collection of searchable on-line papers. In addition to the papers, it provides links to the references and information on similar papers.

For writing tips:

- Consider looking at: Tips for Writing Technical Papers from Jennifer Widom, Stanford University.