EE 126 Project – Spring 2017

For the project this semester, you will explore applications of EE 126 toward either **PageRank** or **Catan**, trailers for which you've seen in lab. We also have an additional project theme, **Digital Communication**, that you can choose.

1 Logistics

This project is worth 5% of your course grade. You should work in groups of 2 or 3. Only one student in each team is to submit the files to bCourses.

Each team should only work on one project. Depending on the project you choose, there will be a task for you to finish in one week. Your final submission will be due the following week.

1.1 Timeline

- February 25: Projects are released
- March 3 (9:00am): Task 1 due on bCourses (20%)
- March 16 (9:00am): Entire project due on bCourses (80%)

1.2 Extra Credit

The top team in each category will be awarded bonus points.

2 Catan

We found a way to simulate a (simplified) game of Catan[1]! Awesome! To ensure your domination on the island of Catan, you will need to design an action policy to win in the fewest number of turns possible, finding optimal settlement locations and determining the tradeoff between building additional settlements versus development cards. To start this project, download the Catan trailer lab (`Project_Catan_Trailer`) and our implementation of Catan (`catan.py`).

2.1 Task 1

Write up a proposal which describes your proposed strategy and at least 2 extra features you want to implement. Follow the instructions in the Catan lab to get acquainted with the platform. There is some starter code at the end that you might find useful.

2.2 Task 2

This task is composed of three smaller parts. You should turn in `Project_Catan.ipynb` along with your report and any additional files your implementation requires.

2.2.1 Optimal strategy

Now you need to minimize the expected winning time by developing a better strategy for the **provided** model of Catan. With this new strategy (implemented as the action function in the `Lab_Catan.ipynb`), run 10,000 simulations of the game and provide the mean number of steps it takes you to win. You should follow the rules!

2.2.2 Extend the model

As you probably have noticed, we cut several corners while coming up with our model of the game. In order to be able to use the strategy in real life, you have to increase the complexity of the model. If you'd like, you can implement some of the actual game rules [2] or you can come up with your own. Some examples of features that are waiting to be implemented:

- The actual game uses hexagonal tiles instead of square ones.
- Rolling a 7 summons the robber, who forces every player to discard half of their resources in the case that they possess more than x (choose x) resources in total. But who gets to choose what to discard?
- Incorporate multiple different kinds of development cards
- Catan is no fun playing by yourself. Could we simulate a game with multiple players (where every player has her own strategy)?

If you decide to implement the existing rules, try to come as close to the actual game as possible! If you invent a new rule that would make the game more interesting, you're free to implement that too!

[1] TA contact: Tavor

[2] which can be found at `http://www.catan.com/service/game-rules`

2.2.3 Report

At the end, please write up a two to three page report that details your assumptions, attempted strategies, motivations behind these strategies, and how you ended up developing your final strategy. At the end, please talk about the extra features you implemented, the implementation details, and how these changes would impact your strategies, and more broadly, the flow of the game.

2.3 Grading

We will generate 10 random boards (warning: these boards may not have all possible resources) of size 4x4 and then run your policy on each board 10 times. Your score will be the average time it took you to win across all boards.

2.4 Submission

Please submit both the report and your software implementation in a zip archive to bCourses.

3 PageRank

Now that you understand the idea that led to the success of the PageRank algorithm, we want you to explore other applications of this powerful tool[3].

3.1 Task 1: Proposal

Come up with a new application for PageRank or Markov chains. Some examples in the past have included:

- rank professors in a more advanced way than we did
- rank another topic – subreddits, tech companies, politicians, etc.
- something different – random walk song generation, hitting time, etc.

See files uploaded on bCourses for additional examples.

Please specify: what is the motivation behind your idea? What is the dataset you will be exploring/analyzing? How do you plan on applying PageRank? What do you expect to see from the results?

The proposal should be less than one page in length and include the names of all team members.

3.2 Task 2: Code and Report

Include all source code you wrote in a zip folder in your submission. Make sure to cite any code you may have copied from the web. You may use any programming language; Python is not necessary.

Summarize your results in a 3-page to 5-page report (preferably typeset in LaTeX). An example structure to the report may include the following sections:

- introduction
- methods (theory and pseudocode)
- experiments
- results/analysis (with figures)
- discussion/limitations

3.3 Grading

The proposal will be graded on detail and clarity.

For the final check-off, a score of 8 out of 10 reflects a good understanding of the algorithm and implementation. A score of 10 out of 10 will require some creativity or innovation.

[3] TA contact: Tavor, David, Andrew, and Kabir

4 Digital Communication

Our last project theme is digital communication[4], and for this project, you will build a system from scratch to transmit a text file between two laptops, using only the speakers and microphones of the computers. Your system should be able to transfer the file at a reasonable speed and it should also be robust to noise, both ambient noise and burst noise (like clapping). For full credit on the transfer speed aspect of the project, you will need to achieve a bitrate of at least 100 bits per second.

4.1 Task 1: Proposal

Your proposal should describe how you plan on implementing your digital communication system. You should also describe how your system will achieve the goals listed above (transfer speed and robustness to noise). The proposal should be about one to two pages in length, and remember to include the names of all team members.

4.2 Task 2: Code and Final Report

You will be building this system from scratch, so feel free to use any programming language of your choice. Please be sure to cite code snippets that you copied from other resources. Include all of your source code in a zip folder in your submission to bCourses.

In addition, write a two to three page report on the project. In the report, you should include a description of your final implementation, some challenges you faced during the project and how you resolved them, and what you learned from the project.

4.3 Grading

The written aspects the project, the proposal and the final report, will be graded on detail and clarity.

To grade your implementation, you will schedule a time to demonstrate your system to one or two members of the course staff. We will give you a text file (.txt format), and you will show us how your system transmits the file from one laptop to another. For the purposes of the competition, we will consider things like how fast of a bitrate you can achieve, how robust the system is to noise, and the creativeness of your implementation.

[4] TA contact: David and Andrew