EECS 126 Project – Fall 2018

For the project this semester, you will explore applications of EECS 126 toward either **Catan**, **Markov Chain Monte Carlo & PageRank**, **Financial Modeling and Prediction** or **Digital Communication**.

1 Logistics

This project is worth 5% of your course grade. You must work in groups of 3 or 4, and are encouraged to use Piazza's teammate finding feature if you are lacking group members. We strongly recommend forming teams as soon as possible. Only one student in each team is to submit the files to Gradescope. If your proposed idea isn't in line with what we (the staff) want, we will let you know shortly after the proposal due date, but this is rare. However, you are strongly encouraged to talk to the staff members handling your category of projects about your ideas before submitting.

Each team should only work on one project. You have 5 days to form a team and decide on a topic, after which you have to submit a Google Form informing us about your team and the topic you will be working on. Over the next week, you must formulate a concrete proposal in line with your topic, and this is due by October 26. Each topic has different requirements for the proposal, and these are listed under the topics in more detail. Your final submission will be due November 18.

1.1 Timeline

- October 15: Projects are released
- October 19: Task 1 Form Due (5%)
- October 26 (9:59 AM): Proposal due on Gradescope (15%)
- November 18 (11:59 PM): Entire project due on Gradescope (80%)

1.2 Extra Credit

Standout projects 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.ipynb`) and our implementation of Catan (`catan.py`).

2.1 Task 1

Write up a proposal which describes your proposed strategy. 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

2.2.1 Optimal strategy

You should come up with a strategy to minimize the expected winning time by developing a better strategy for the **provided** model of Catan. You are responsible for creating a new strategy (implemented as an action function, takes in 1 argument, the player), a planBoard function, that takes in the board and can be used to produce an overall plan for all trials run on that boards, and a dumpPolicy which determines what resources you will drop if you have too resources and a 7 is rolled. You are only allowed to modify these three methods, and your action / planBoard / dumpPolicy functions must operate within a time constraint (under 3 minutes for 100 trials on a fixed board). Make sure to follow the rules of Catan.
We would also like you to make a separate (or the same) policy optimized for playing against other players on a single board. This will require you to consider potential adversarial strategies. We will provide a staff bot later on for you to test your strategy against.

We would also like you to make a separate (or the same) policy optimized for playing against other players on a single board. This will require you to consider potential adversarial strategies. We will provide a staff bot later on for you to test your strategy against.

Your planBoard, action, and dumpPolicy methods should be submitted in a separate file `catanAction.py` (see sample), along with all relevant helper methods. This means that you should not put any of your work in `catan.py`, as we will be using a plain `catanAction.py` when we test your action method.

2.2.2 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 this point, please include the mean time your algorithm takes to win, averaged over 10 randomly generated boards, running 100 trials per board.

[1] TA contact: Eric, Justin, Katie

2.3 Grading

We will generate 10 boards (warning: these boards may not have all possible resources) of size 4x4 and then run your policy on each board 100 times. Your score will be the average time it took you to win across all boards. In addition, we will be running student submissions for multiplayer strategies against each other and announcing the top teams.

2.4 Submission

Please submit `catanAction.py` and your report to Gradescope.

3 Markov Chain Monte Carlo & PageRank

Now that you understand the idea that led to the success of the PageRank algorithm, and saw in lab how MCMC could be used to sample from distributions and solve ciphers, we want you to explore other applications of these powerful tools.[2].

3.1 Task 1: Proposal

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

- model events using a Hierarchical Bayesian model and estimate the parameters using MCMC
- Perform special cases of MCMC (such as Gibbs sampling) for parameter estimation over a hierarchical model for a real world dataset
- sample from the posterior distribution given the data of the weights of a neural network using MCMC instead of gradient descent to find the optimal solution
- 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 the course website 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 or MCMC? 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

[2]TA contact: Kurtland, Avishek, Payam, Amay, Adarsh, Kanaad

3.3 Grading

The proposal will be graded on detail and clarity.

For the final check-off, a score of 6 out of 10 reflects a good understanding of the algorithm and implementation. A score higher than 6 out of 10 will require some creativity or innovation.

4 Financial Modeling and Prediction

One of the most common domains which uses probablistic modeling is that of financial prediction and estimation. Having seen a variety of modeling ideas in class, this project[3] will have you develop models over real-world financial stock and commodities data from the American markets, and predict the progression of these prices over time. Your models are expected to be reasonably complicated to incorporate macro-economic ideas, while allowing for robust probabilistic modeling. There is a lot of academic work in this area (a quick search over Google Scholar for "financial modeling" will reveal a lot!), and you should find enough past research to provide a solid starting point.

4.1 Task 1: Proposal

Come up with modeling techniques for "training" data provided in the following format - Time indexed (over a common time frame) lists of prices for numerous financial instruments traded on the floor of the NYSE, NASDAQ, NYSE Amex Equities, BATS, etc. The required part of the project is to model US Equity prices across companies and sectors, with there being an optional bonus component to extend your model to include pricing of commodities, bonds, options and other derivatives. There is another independent bonus part to train your models over the training data, and then predict across a common "test" set, and measure predictions.

Some starting points to do this include:

- Creating parametric (possibly hierarchical Bayesian) models for the evolution of stock price over time
- Developing a multi-stage Hidden Markov Model for the price evolution
- Analyzing stock prices as signals in the frequency domain
- Ensemble techniques combining above methods
- Extending ML techniques (multistage regression, classification, feature engineering, kernelized SVM-like techniques, etc) to this dataset. Note: Using neural methods for prediction are good for comparison, but are inadmissible for this project unless you can provide variance bounds for your predictions.

Please specify: What is the motivation behind your idea? What do you expect to see from the results?

The proposal should be less than two pages in length and 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 (use of Jupyter Notebooks and Python is strongly encouraged though). Please be sure to cite code snippets that you copied from other resources, and cite any academic work you build off of. Include all of your source code in a zip folder in your submission to Gradescope.

[3] TA contact: Nikunj

In addition, write a three to five 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. Also explicitly list out the technical outline of the project, and the specific results observed from various modeling strategies.

4.3 Grading

The written aspects the project, the proposal and the final report, will be graded on detail and clarity, as well as technical prowess and accuracy.

4.4 Bonus 1

Extend your modeling techniques from US Equities to the various market instruments listed above, and present accurate findings.

4.5 Bonus 2

There will be a test-dataset provided (following the same format as the training set). Do not train over the set, but simply predict the prices, and report your accuracy and goodness-of-fit. The most accurate models will be awarded glory!

5 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.

5.1 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.

5.2 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 Gradescope.

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.

5.3 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 and how robust the system is to noise.

[4]TA contact: Ray, , William, Raghav