Aditya Pidaparti
Progress Report Draft 1
1/24/18

Fall 2016 - Independent Study

- Objective:
    - Learn about machine learning, complete papers, and setup* for a UROP in the spring
    - *Setting up implied finishing complete porting of MATLAB code to Python.
- Results
    - Completed Coursera Machine Learning course
    - Submitted reports on 2 papers
    - OLU code in Python was working in most cases but went to 0 wealth unexpectedly with a few initial parameters.
- Successfully ported code to work in most (80%+) by mid-October, but then got stuck on a bug where in some cases the Python wealth would vary greatly from MATLAB result. Typically, the Python code would have a wealth going to zero while MATLAB would not. Trying to debug this issue will take another 3 months.
- In early December, suffered a head injury in a bike accident on the way home, smacked head right into pavement. Resulted in an Incomplete for the class, with all required class materials completed *except for the final report on the research*.

Spring 2017 – UROP (plan summary at bottom)

- Objective:
    - Finish porting code from MATLAB to Python
    - Do UROP, which was to modify the OLU algorithm by inclusion of risk.
- Results:
    - Finished debugging Python code completely. All tested parameters resulted in equal wealth return in both MATLAB and Python, given some floating-point error uncertainty.
    - Developed methodology and code to find betas of stocks. Also had to find an index (S&P500) that could be used and collected data going back to the dates of the dataset.
        - Had to account for differences in trading vs. calendar days and built code that is reusable to find betas given an index dataset.
    - Built the portfolio rejection strategy. However, the method created had bugs that were not discovered until Dec 2017 (detailed below).
- Spent most of the Spring trying to find the bug in the code. Continued working and discovered it was simply an indexing error between MATLAB and Python in the find_y function, as MATLAB begins with an index of 1 and Python with 0.
- Collected data to find betas. The overall development of beta computation is developed an Jupyter notebook
    - To summarize how it was found: Collect index data and clean it, then find the correct date ranges, next adjust for trading vs. calendar days, and then calculate betas over varying date ranges (to account for short and long-term volatility).
- Built reject method, was not able to test quickly as had 12,000 test cases to complete.
- Filed an extension in April as I would need additional time to have this completed.

Summer 2017 - Summer Internship

- Objective: Continue working on UROP as much as possible
- Results: Not much was achieved. Internship was more time-consuming than expected.
- Began trying to find results. Tried to run on personal laptop but would take too long (~80 days) to run all 12,000 test parameters, testing across all the previous parameters in addition to multiple beta ranges.

Fall 2017 - Data collection (failed)

- Objective: Finish UROP
- Results:
    - Collected data on all 12,000 cases, only to realize my data collection methods were poor, and ended up actually producing poor results by mistake. Only made the realization over winter break.
- The semester was so stressful and busy I withdrew from 2 of my classes.
- Began taking time to run results manually. Entered discussion with CSE IT department about using machines to collect data, was denied. Nick recommended to run OLU across shorter timespans but it was felt that was an inadequate way to compare results, so began the tedious process of manually running OLU.
- Manually ran reject NYSE algorithm on UMN machines. However, a bug in code resulted in faulty results.
    - CSE Unix machines interpret "python" as Python 2.7, while my computer recognizes "python" as Python 3. Python 2 does not do automatic int to float conversions. This resulted in my list comprehensions for initial parameter generation rounding, so a value of eta=0.05 became eta=0.
    - Python 3: 1/10 = 0.1 . Python 2: 1/10 = 0.
    - Made the above realization after end of semester.

Spring 2018 - Finish everything

- Objective: Finish UROP, turn in final progress report and UROP paper.
- Results: In progress.

    Re-read OLU paper and multiple background texts on ADMM/Lagrangians to understand better how ADMM and the code works.

    - Developed extrema tests to run. Discovered that the unrestricted portfolio beta only varies from about 0.4 to 1.8, compared to an index value of 1.
    - Currently have built a testing suite for the NYSE dataset doing random sampling (with replacement, so I can parallelize the process). Good results appear to be coming out.
    - Currently running the tests “under the radar” on the UMN Linux machines, by manually running the tests via SSH and Screen.
- Developed and finished building a new reject method. As testing for all parameters took too long, did random sampling tests instead of comprehensive ones. Parallelized and ran across CSE machines. Have gotten back initial results and looking at them. It appears that limitation of risk results in regression towards a mean value of 26.8x return.

Current issues

- I have no idea how to implement an 'optimize' method where I can attempt to improve the algorithm's selection based on an update step in spare_port_admm.py. I have looked at Boyd's papers, looked at his lessons, read multiple papers, and I am still stuck. I was hoping to have a completed work to turn in, but I don't.

- When I deal with any concepts like a simplex or L1 ball, I can handle them individually, but applying them together to make the correct mathematical formulation is beyond me.
- I am wondering if it’s worth it to run tests on the S&P500 dataset as well. I want to get your input on this, as I’ve stretched this study out well beyond what I should have.

***UROP plan summary**

- The UROP plan was to develop OLU code further by inclusion of risk. The prediction was that inclusion of risk restraint measures would results in portfolios moving less, and so staying closer to a buy and hold strategy (26.8x return in the NYSE datset)
- The planned modification was to use the beta as a measure of riskiness, as it is conveniently tied to a regularized value of 1.
- Two strategies were planned. The first was a “rejection” method, in which the portfolio suggested by OLU was created, and if it had too high of a beta, the previous day’s beta was used instead. The other was an “optimize” method, where projections using ADMM would occur to see if we could optimize for the riskiness of individual stocks by weighting them less heavily.