

Hudson & Thames

March apprenticeship writeup

Franz Krekeler

16 February, 2020

Outline

1. Setup
2. Way of work
3. Difficulties & Learnings
4. Outlook

Setup

After reading the paper I started prototyping in a Google Collabotory.

At first I implemented each step very carefully. After some time I went on defining the basic functions and started coding in Visual Studio Code. After finishing most functions I started optimizing them and trying to go for as much vectorization.

```
[ ] 1 %%time
2 all_combinations = []
3 #for name,group in target_df.groupby('l0_codes'):
4 for name,group in target_df.groupby('level_0'):
5     combinations = group.level_1.tolist()
6     #combinations = group.l1_codes.tolist()
7     high_score = 0
8     high_list = []
9     for comb in itertools.combinations(combinations,3):
10         comb_a = [name] + list(comb)
11         score = df_corr.loc[comb_a,comb_a].sum().sum()
12         if score > high_score:
13             high_score = score
14             high_list = comb_a
15     all_combinations.append({"stocks":high_list,"score":score})
16     print(len(all_combinations))
17     break
```

```
1
CPU times: user 27.3 s, sys: 367 ms, total: 27.7 s
Wall time: 27.1 s
```

```
[ ] 1 %%time
2 #Traditional approach
3 all_combinations = []
4 #for name,group in target_df.groupby('l0_codes'):
5 for name,group in target_df.groupby('level_0'):
6     #%prun
7     all_combinations.append(get_best_pair(name,group))
8     break
```

```
CPU times: user 287 ms, sys: 244 ms, total: 531 ms
Wall time: 648 ms
```

First 100x increase of speed in the traditional approach



Way of work

I started by writing down the most important notes from the paper in the Jupyter notebook. If I didn't understand something I tried to test and calculate it with a small amount and then develop further. For visualization I was aiming for using plotly express / dash.

After getting the class ready I started to work with inheritance. My idea was to divide the modules and make them accessible like modules H&T or other popular libraries.

Difficulties & Learnings

1. Beginning: Understand what to do.
2. Middle: Understand how to do it.
3. Ending: Wrapping everything up under time pressure.

In the beginning I was trying to understand what would be the most important and my mind was especially on the design and architecture side.

After understanding the algorithms are most important, I started thinking about how to best implement them.


In the end I had a bit of testing, staging, writing phase where I developed something in a notebook and then added it to the modules. Especially with time pressure I was more glued to code, even though I think the writing was at least as important.

Mistakes:

- Getting glued too much into a certain line of code instead of concentrating on the look and feel
- Not working in VS Code from the start as many of my notes got lost in the notebook.

Learnings:

I learned a lot, especially since copulas is something I have never worked before. I especially was thankful for the helping and clear instructions in the GitHub repository of H&D.



I had so much fun, that even though I had fully booked week, I spent every free minute on the challenge. I learnt a lot about speed optimizing, Einstein sums and working in multiple dimensions. I definitely can learn a bit more about how to implement formulas in a clean way.

Also I notice the more I tried to explain the problems to friends the better I understood the tasks

Outlook

I would like to add more visualization especially some density graphs of the copulas.

In the documentation the formulas need to be more explained.

Further I would like to maybe even further improve the speed, as I think this could be relevant for backtesting.

Most importantly I'm grateful for the challenge.

