Milestone 2

A. Einarsson, B. Coremans, G. Palsson, S. G. Jonsson,

Datasets

A few stock market datasets were discussed but in the end the stock sample dataset from 2018 was chosen because of its size and structure. This dataset contains information on more than 2000 stocks from various markets in the form of `.txt` files. It features the name of the stock, date and time of update, the first-, highest-, lowest-, closing price, sum of volume of all transactions, all within this update frequency. The group decided to do pairwise correlation measures between four quarters. Each quarter contained 83 days, so the length of each stock vector was 83.

The frequency of updates is inconsistent within and between stocks, some update every minute, some once a day and some somewhere in between. First the `.txt` files where merged, a `.csv` header added, and a column containing the name of each stock was added. This yielded one big `.txt` file, roughly 9 GB, containing all the stock data sorted by the name of the stock. Having one big data file made it easier to upload online and work on collaboratively. The file was split up into dataframes, one for each stock. The dataframes had varying sizes because of the inconsistency of update frequency between stocks. A decision was made to use one measurement a day for all the stocks. This means we throw away every line except for the last update for each day. This was done because a good proportion of the stocks only had fewer than 5 measures per day and instead of interpolating and making up thousands of minutes for them we simply treat the data as a daily update. The minute wise updates are not so drastic that this should affect the overall trend in the pairwise correlation. The missing days were interpolated (linearly) for all stocks and cut off some days at the beginning and end of the year. This was done because not all stocks started updating on January 2nd all through December 31st. This resulted in a dataset that started on the 4th of January and ended on the 27th of December. Those roughly 20 stocks that did not meet this requirement were thrown out. For each stock, four names were produced for each quarter.

Correlation Measures

Total Correlation is one of several generalizations of mutual information and is also known as the multivariate constraint or multi-information. It quantifies the redundancy or dependency among a set of n random variables.

For a given set of n random variables $\{X_1, X_2, ..., X_n\}$, the total correlation $C(X_1, X_2, .., X_n)$ is defined as the Kullback-Leibler divergence from the joint distribution $p(X_1, ..., X_n)$ to the independent distribution of $p(X_1)p(X_2)\cdots p(X_n)$,

$$C(X_1, X_2, .., X_n) \equiv D_{KL}[p_(X_1, .., X_n)||p(X_1)p(X_2)\cdots p(X_n)] \tag{1}$$

This divergence reduces to the simpler difference of entropies

$$C(X_1, X_2, ..., X_n) = \left[\sum_{i=1}^{n} H(X_i)\right] - H(X_1, X_2, ..., X_n) \tag{2}$$

where $H(X_i)$ is the information entropy of variable X_i, and $H(X_1, X_2, ..., X_n)$ is the joint entropy of the variable set $\{X_1, X_2, ..., X_n\}$. In terms of the discrete probability distributions on variables $\{X_1, X_2, .., X_n\}$, the total correlation is given by

$$C(X_1, X_2, .., X_n) = \sum_{x_1 \in \mathcal{X}} \sum_{x_2 \in \mathcal{X}_2} \cdots \sum_{x_n \in \mathcal{X}_n} p(x_1, x_2, .., x_n) \log \frac{p(x_1, x_2, ..., x_n)}{p(x_1)p(x_2)\cdots p(x_n)} \tag{3}$$

The Pearson correlation is a statistic that measures linear correlation between two variables X and Y. It has a value between $+1$ and -1, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation.

Pearson's correlation coefficient when applied to sample is often represented by r_{xy}. Given paired data, in our case one stock and the average of two or more stocks:

$$\{(x_1, y_1), ..., (x_n, y_n)\}$$

where n is the number of stocks, r_{xy} is defined as:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4}$$

In Milestone 1, we considered Mutual Information. Total Correlation reduces to that measure when $p = 2$. Total Correlation is a "distance" between two or more probability distributions whereas Pearson Correlation is a linear distance between two random variables.

System Architecture

First the number of stocks for comparison is selected and the stocks are selected at random from the dataset. This is done because the stocks are sorted by stock markets and running comparisons on the full 1700 stocks is to computationally heavy. Getting a subset of stocks from various stock markets is desired so a random approach was chosen. The stocks are collected on the form `<name,[time series]>`. After initializing Spark and defining the aggregation and correlation functions, a function named `compute_correlation()` is called, corresponding to `correlationFunction` from the milestone description constraints. It takes as input, the stocks to be compared, aggregation function, correlation function, p (size of multiple correlation) and partition (how much work each Spark worker is given).

To make the desired combinations of stocks the function `get_subsets()` was used. The function is described above and is based on bitwise subset generation. The type of subsets to generate depends on input variable p and the correlation function. When the correlation function is chosen as total correlation the subsets to generate are all unique unordered combinations of the stocks of size p. For this we use a function similar to `get_subsets()` pseudocode above, with a few adjustments. When the correlation function is chosen as Pearson correlation all combinations of stocks of sizes 1 to $p-1$ were generated.

Distribution of Computation

The aggregation calculations were distributed to Spark workers. Each array containing every subset of a specific size was parallelized and flatmap applied using the aggregation function. Thus each Spark worker got one subset to aggregate at a time and returned the combined time series.

When generating the keys and splitting the subsets into partitions no Spark feature was used but it still takes a trivial amount of time. Two subsets of combined length p were given the same key. To refresh, the subsets contain stocks. This means if $p = 5$, all combinations of subsets of length 1 and 5 were given the same keys, similarly subsets of length 2 and 3 were given the same keys.

The list containing the partitioned data was parallelized. Given number N of `<key1, name1>`, `<key1, name2>` different pairs to compare, they are split into M number of partitions. This means that each partition contains N/M number of pairs, given N is the number of desired comparisons. During experimentation it was noted that at around $M = 1000 - 3000$ it reached its optimum, although after $M = 100$ the difference was small. It did not seem to matter much for M what the size of N was. The spark map function initializes M many workers instead of N if reduce by key would have been used on the pairs without partitioning. Each worker is responsible to calculate N/M comparisons and does so using either reduce for Pearson or map for Total correlation. This part is the bottleneck of the process, not surprisingly, because here are most of the calculations carried out.

Theoretical Complexity and Performance

The following computers and servers were used for the computational part

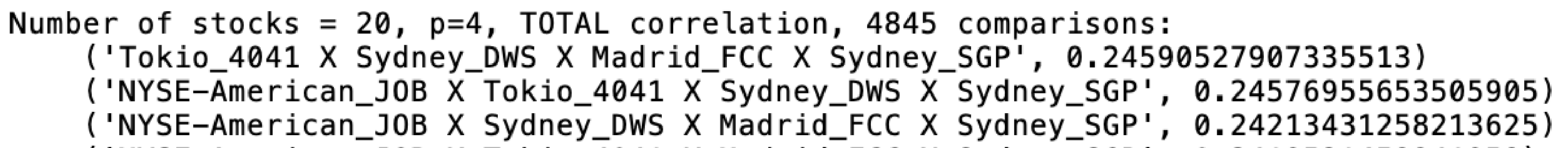
- MacBook Pro from 2015. Processor is 2,2 GHz Intel Core i7, and RAM 16 GB 1600 MHz DDR3.
- MacBook Pro from 2019. Processor is 2,2 GHz Intel 8-Core i9, and RAM 32 GB 2400 MHz DDR4.
- Server specs: Amazon web service cluster that contains three instances(1 master and 2 slaves)

The calculation of the correlation measures vary vastly in speed. Generating the subsets for both is relatively fast but the calculations of Total correlation is much slower then Pearson. Although there are fewer comparisons and the work is distributed in the same way, Total correlation is always slower to make those calculations. This is probably due to the fact how they are calculated on the workers but not how they are distributed. Therefore when comparing the two methods on the same set of stocks, Total correlation is the always bottleneck.

The subset generation has different computational complexity for the two correlation measures. This is because Total correlation only generates subsets of size p but Pearson generates all subsets of size less than p. The number of subsets generated then reflects the number of comparisons made. The number of calculations for Pearson is higher than Total Correlation. And the gap between the two measures grows rapidly after taking more than 100 stocks from the data.

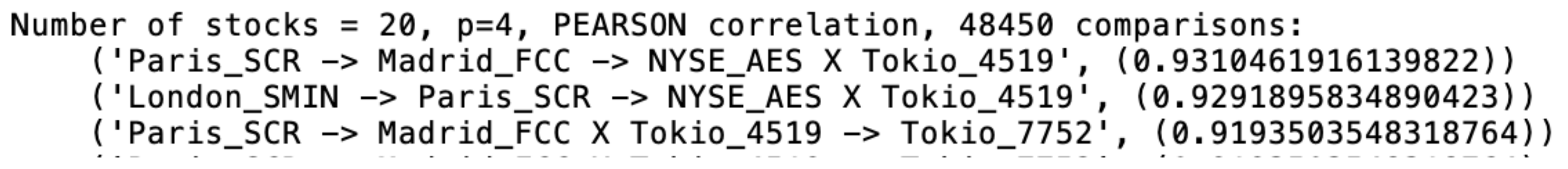
Insights

It is interesting to compare the results from the Pearson correlation and Total Correlation. The Pearson Correlation computes the linear correlation between the vectors while the Total Correlation measure is more general and measures the redundancy or dependency among a set of n random variables. Pearson Correlation assumes for linearity and the absence of outliers. The assumption of linearity is perhaps not accurate when looking at stock data. And if there are outliers in the stock data, some Pearson Correlation measures are invalid. Therefore, it might seem more reasonable to emphasize more on the Total Correlations which does not have as many assumptions.



```
Number of stocks = 20, p=4, TOTAL correlation, 4845 comparisons:
    ('Tokio_4041 X Sydney_DWS X Madrid_FCC X Sydney_SGP', 0.24590527907335513)
    ('NYSE-American_JOB X Tokio_4041 X Sydney_DWS X Sydney_SGP', 0.24576955653505905)
    ('NYSE-American_JOB X Sydney_DWS X Madrid_FCC X Sydney_SGP', 0.24213431258213625)
```

Figure 1: Total Correlation results for $p = 4$



```
Number of stocks = 20, p=4, PEARSON correlation, 48450 comparisons:
    ('Paris_SCR -> Madrid_FCC -> NYSE_AES X Tokio_4519', (0.9310461916139822))
    ('London_SMIN -> Paris_SCR -> NYSE_AES X Tokio_4519', (0.9291895834890423))
    ('Paris_SCR -> Madrid_FCC X Tokio_4519 -> Tokio_7752', (0.9193503548318764))
```

Figure 2: Pearson Correlation results for $p = 4$

The Total and Pearson Correlations from Figure 1 & 2 are very different. There are no stocks in common for the two measurements. The assumptions of Pearson Correlation is perhaps violated and non-linearity is too extreme.

Video

Link to video: `https://www.youtube.com/watch?v=U8-ebZW6e2s&feature=youtu.be`