Financial Big Data Project[*]

Mengjie Zhao
mengjie.zhao@epfl.ch

Lu Zhong
lu.zhong@epfl.ch

ABSTRACT

To acquire a big picture of the stylized facts of financial big data, we, at the first step, collect the huge amount of data of U.S. stocks and commodities future over a relatively long period and analyze these data to test and verify the characteristics provided in *”Empirical properties of asset returns: stylized facts and statistical issues”*. Acting as a portfolio management, we need to choose assets from huge number of assets U.S. Stock. In order to deal with big data, we use a technique backed up by random matrix theory to filter the assets of U.S. stocks that would compose the portfolio. After that, we apply mean-variance strategy to find optimal weights that will minimize the variance,i.e. the risk, of portfolio with given expected return. Furthermore, we evaluate the performance of our strategy by regression on the $S\&P500$ benchmark.

1. INTRODUCTION

Since the financial markets have undergone rapid development in the past decades owing to technological advantages, huge data has been produced and kept by financial sectors such as banks, asset management firms and even industry companies. These data share the same characteristic of big size (volume), complexity (variety), and rate of growth (velocity), which make them difficult to be captured, managed, processed, and analyzed by conventional technologies and tools. Thus, how to process big data with advanced methods and tools is popular and needs to be improved.

Unlike traditional datasets, which collect data from specified sources during specific time periods or constraints, big datasets are constantly growing and updated and may reach multiple terabytes. Compared to traditional datasets, the analytics entail more data extraction and manipulation to ensure the data can be translated into something that is usable and comprehensible. In this project, we use huge assets from US stock market to find reliable assets to construct our own portfolio.

2. DATA CLEANING

Data cleaning and preprocessing are the key steps before carrying out any further experiments and analysis. In this section, we firstly provide some background information about datasets we used during this project. After that, we demonstrate how we cope with NA and outliers among the raw datasets, which is the key step in cleaning and preprocessing provided datasets. We provide two tables summarizing the properties of raw and processed datasets.

2.1 Datasets description

We carry out our experiments and analysis on **two** datasets, namely the US stock price dataset (USSP) provided by Prof. Damien, and one commodities price dataset (COMP) collected by the authors from **Quandl** and **Yahoo! Finance**. Following table gives a summary of the two raw datasets.

Table 1: A summary of raw datasets

Dataset Name	USSP	COMP
Start Date	1950-01-03	2000-01-01
End Date	2014-03-18	2016-11-30
Num. Timesteps	16291	4334
Num. Stocks/Coms	1053	10

2.2 Dealing with NAs

NA value is a crucial factor that has to be carefully considered and treated. Strategies of dealing NA can generate significant impacts on the performance of constructed portfolios. We use several approaches to deal with NA value, details are as follows:

- For days that all stocks have NA value, we simply remove that day from our dataset, as during these days stock exchanges may closed.
- For stocks or securities that have NA value for all days, we remove that stock. This may comes from others stock exchanges outside us.
- When calculating correlation/covariance matrix, we ignore all rows/columns that contain any NA value. This procedure is adaptive, especially in the testing procedures. However, these data will not be deleted from our dataset.

[*] This work is under the guidance of Prof. Damien Challet.

Based on the our NA dealing strategy, we can control the quality of data to some extend, and then use these filtered data to carry out further experiments.

2.3 Dealing with outliers

Outlier is an observation point that is distant from other observations. As some models are sensitive to outlier, it is reasonable to remove them from the dataset, or tame them back to normal ranged value. In our settings, we treat the securities that have log return larger than 0.2 are outliers. Hence, we set log returns as $min(0.2, LogReturn)$. In addition, for USSP dataset, we only consider data after the year of 1995 as we believe that newer data can better reflect the trends of the financial market. Following table summaries the processed datasets.

Table 2: A summary of cleaned datasets

Dataset Name	USSP	COMP
Start Date	1995-01-04	2000-01-03
End Date	2013-09-11	2016-11-30
Num. Timesteps	4706	3701
Num. Stocks/Coms	1034	10

3. DATASETS' STATISTICAL FEATURES - STYLIZED FACT ANALYSIS

In this section, we evaluate some statistical properties of the USSP and COMP datasets. Similar to other type of datasets, it is extremely easy to explore these properties like mean of log returns and standard deviations etc. However, for concisions, we will only focus on the special characteristics of financial datasets such as heavy tails and absence of linear correlation, since it is these special traits that distinguishes the financial big datasets from other datasets. In following parts of this section we carry out evaluation of some market stylized facts. To illustrate these features (except **4.3 gain/loss asymmetry** and **4.7 leverage**), we choose the raw data(only wiping out NA) of **MCD** (McDonald's Corporation) stock prices($)and **CORN** future price($).

3.1 Absence of autocorrelations:

Autocorrelations of asset returns are often insignificant for the not very small time scales ($\sim 20min$).Here, the data are based on daily return.

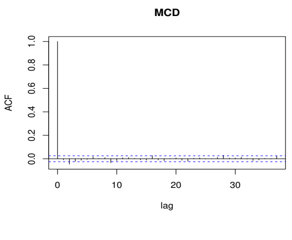


Figure 1
Autocorrelation of MCD

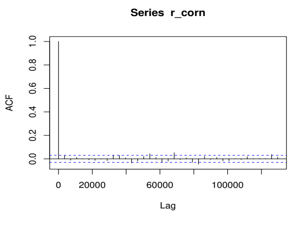


Figure 2
Autocorrelation of CORN

3.2 Heavy tails:

For testing heavy tails, we choose Q-Q plot, which is a plot of the quantiles of our data and normal distribution. One can see that the both MCD return and RICE future return bear heavy tails,i.e. there are more extreme values for return's empirical distribution than normal distribution.

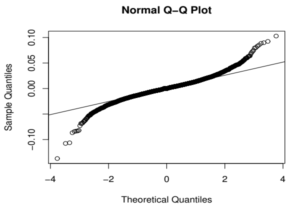


Figure 3
Q-Q plot of MCD

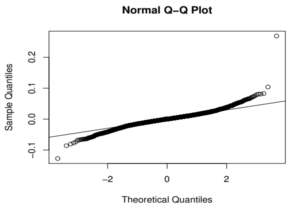


Figure 4
Q-Q plot of CORN

3.3 Gain/loss asymmetry:

In financial market, one observes drawdowns in stock prices and stock index values but not equally large upward movements. To make sure of it, we calculated the skewness of gain/loss asymmetry and plot the distribution of skewness of US stocks and commodities future. As we can see, there are more negative numbers than positive,i.e. negatively skewed of skewness.

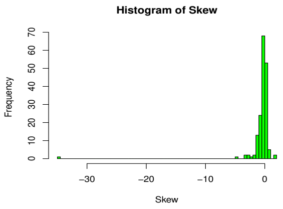


Figure 5
Skewness of US stocks

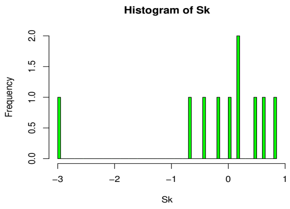


Figure 6
Skewness of commodities

Table 3: Skewness of chosen dataset

Dataset Name	US stocks	Commodities future
Skewness of skewness	-3.7519	-1.5500

3.4 Aggregational Gaussianity:

The larger time scale t over which returns are calculated, the more their distribution looks like a normal (Gaussian) distribution. Using JB test, set $\Delta t = [10 : 40 : 410]$ to calculate the P-value and show them in the following figure.

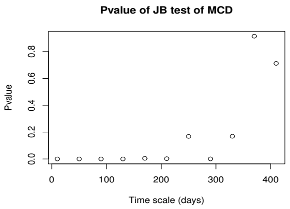


Figure 7
JB test P-value of MCD

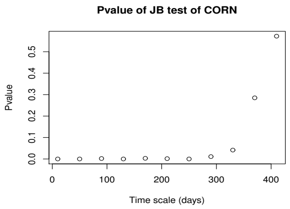


Figure 8
JB test P-value of CORN

3.5 Volatility clustering:

There exist volatility clusters: volatility can be low for certain time periods and then quite large for other periods. Say differently, large variations of price (irrespective of the sign) are expected after a large variation of a price.

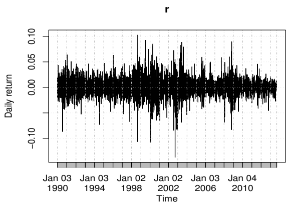


Figure 9
Daily return of the MCD

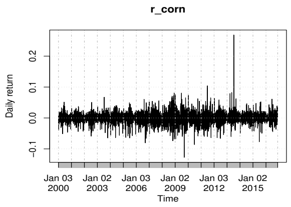


Figure 10
Daily return of the CORN

3.6 Slow decay of autocorrelation in absolute returns:

On contrary to the autocorrelation of return, the absolute return bears slow decay of autocorrelation and thus provides the evidence of serial correlation.

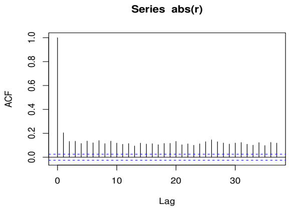


Figure 11
Autocorr. of abs(MCD)

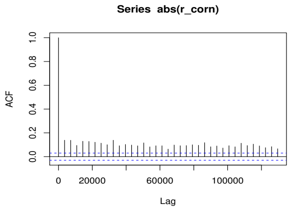


Figure 12
Autocorr. of abs(CORN)

3.7 Leverage effect:

Most measures of volatility of an asset are negatively correlated with the returns of that asset.For verifying this characteristic, we use sliding window of 20 (roughly monthly) and step equal to 1 to calculate every window's expected return and volatility. Thus we then compute the correlation of the return and volatility over the same time period. (The number of commodities are not enough large and hence we only use dataset of US stocks)

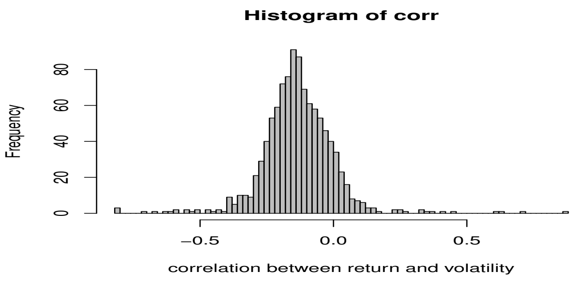


Figure 13
Leverage effect on US stocks

4. COVARIANCE MATRIX PROCESSING

One of the key benchmarks for evaluating performance of a constructed portfolio is to quantify its risk, which is achieved by studying the covariance matrix of different securities. In this section, we show the procedures that we followed in abstracting valuable mathematical information from the covariance matrix.

It should be noticed that we calculate the correlation matrix in a rolling manner, meaning that we have a window scanning the time series and calculating covariance or correlation matrix of returns within that window. Typically the window have size T which is 2 to 3 times of the number of stocks that have none NA value in the dataset.

4.1 Filtering the correlation matrix

In previous section, we obtained time series of log returns of each security. As a result, we can easily calculate the correlation matrix directly via using command `cor`.

For financial engineers, it is crucial to be able to pick up securities to construct a well performed portfolio. It is known that a portfolio constitutes of roughly 50 securities is already good enough to beat the market. As there are 289 stocks in our filtered USSP, we implement a filtering technique to the correlation matrix based on the random matrix theory, to focus on the principle components of the correlation matrix.

For a correlation matrix obtained in one specific rolling window along the time ticks, we implement eigenvalue decomposition and set all $\lambda < \lambda^+$ to zero to suppress the risks generated from random noise. As shown in Figure 14, we only consider $\lambda > \lambda^+$, which corresponds to $\lambda > 2.46$ in this case, and $\lambda_s < \lambda^+$ that corresponds to the peak density in following figure are set to zero when reconstructing the correlation matrix via using formula:

$$C^{(filtered)} = V' \ \Lambda^{(filtered)} \ V \qquad (1)$$

where V the matrix correspond to remained eigenvalues.

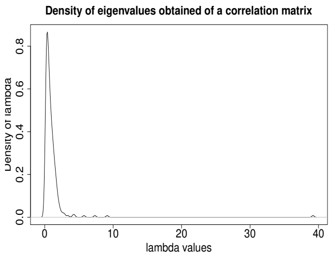


Figure 14: An eigenvalue distribution

After filtering the correlation matrix, we then can recover the covariance matrix using formula (1).

4.2 Shrinking the covariance matrix

The shrinkage procedure, proposed by Ledoit and Wolf, targets to pull the most extreme coefficients towards more central values, thereby systematically reducing estimation error where it matters most. However, it tends to select small eigenvalues, conflicting with the target of random matrix theory, thus we do not include this technique in our project. Nevertheless, the shrinkage can implemented by calling function `cov_shrink` in the package *tawny*.

5. COMMUNITY DETECTION

After obtaining cleaned covariance matrices from the rolling windows, we firstly try to detect communities among the stocks and securities. Detecting the communities can be considered as a clustering procedure. Stocks or securities with similar return patterns would be assigned in same communities.

We mainly utilize the *igraph* package to calculate and visualize communities among stocks and securities. The package provides several available algorithms and we use the fast and greedy minimum spanning tree algorithm to calculate the communities. Following two figures show the communities among the USSP and COMP datasets (for better view, we only show 60 stocks of USSP, for full communities please check appendix).

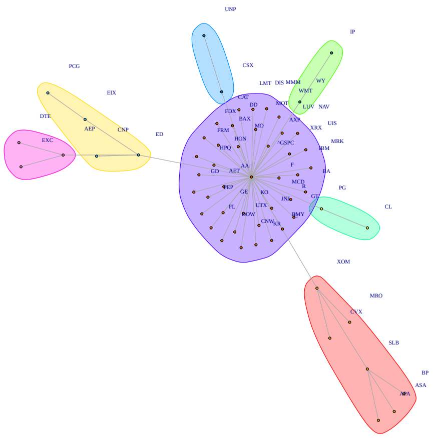


Figure 15: Detect communities among USSP

In above figures, different communities are colored differently. Stocks or securities within the some community have higher covariance hence result in higher risk if all of them are included in the portfolio, which should be avoided in practice.

6. IN-OUT SAMPLE RISK COMPARISON

In a rolling window, we utilize the *portfolio.optim* function in *tseries* package to get the optimal portfolio. After that, we try to test the risky level of the constructed portfolio using out-of-sample data which has same length as the in-sample window. We define the out-of-sample testing data as follows:

- Assuming a rolling window has size T and ranges from $t-T+1$ to t. Then the data within this window is defined as the in-sample data.

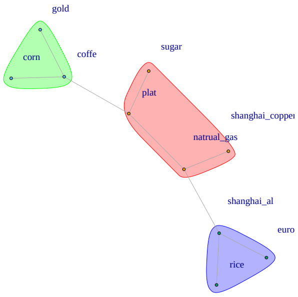


Figure 16: Detect communities among COMP

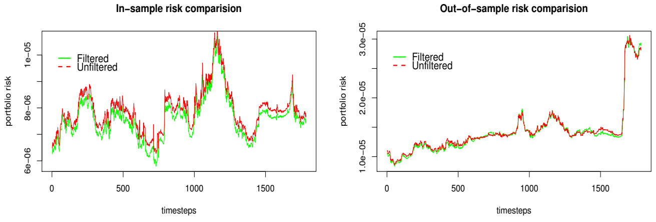
- The corresponding out-of-sample data for this in-sample data is defined as the data within the window ranges from $t+1$ to $t+T$.

In addition, we define the ration between out-of-sample risk and in-sample risk as follows:

$$\Phi = \frac{Risk_{out}}{Risk_{in}} \tag{2}$$

We expect that the ratio Φ normally would be larger than one, as our portfolio risk is optimized with the in-sample data window, hence it is reasonable that the reality (out-of-sample data) would surprise us.

We carry out comparison procedure for both portfolios obtained from filtered and unfiltered covariance matrices from previous sections. We firstly show in-sample and out-of-sample risk of the two portfolio separately, then plot the Φ comparison. Following two figures show the risk performance of the two portfolios (cleaned USSP, 1998-07-29 to 2009-03-27 daily).



(a) In-sample risk (USSP) (b) Out-of-sample risk (USSP)

Figure 17: Portfolio risk comparison for USSP dataset.

Observing the two figures, it can be easily found that the portfolio constructed from the filtered covariance has lower risk than the unfiltered portfolio, which meets our expectation. Another interest result is that the risky level of both portfolio is very high during the time period from year 2008 to 2009, which may comes from the severe financial crisis during that time.

Sequentially, we can plot Φ based on data in above two figures. As shown in following Figure 18 (cleaned USSP, 1998-07-29 to 2009-03-27 daily):

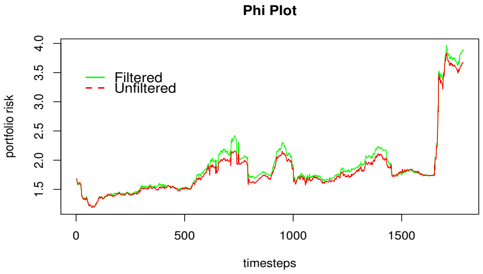


Figure 18: An eigenvalue distribution

Observing the figure, we can conclude that:

- Φ_{fi} and Φ_{unfi} are both larger than 1, which meet our expectations – the risk in the in-sample window is minimized in the portfolio optimization procedure, but the the out-of-sample risk should be large, as we focused on fitting on the in-sample data.
- It can be observed that in most timesteps $\Phi_{fi} > \Phi_{unfi}$.
- Given in both in-sample and out-of-sample cases, risk of filtered portfolio is smaller than unfiltered portfolio while $\Phi_{fi} > \Phi_{unfi}$ we can safely say that the filtering procedure introduced by the random matrix theory can reduce the portfolio risk effectively.

7. PERFORMANCE

Up to now, we build our mean-variance portfolio strategy by minimizing the variance over the given length of the test windows T which in reality is comparable to the investment horizon T. Applying this method, we up to now have already computed the weight vectors over $[t_i - T + 1, t_i]$ of US stocks for each test window and thus we can also calculate the variance, as we did in the previous section, and expected return over the next investment horizon $[t_i + 1, t_i + T]$. To evaluate the performance of our mean-variance strategy, we can calculate the fitted sharp ratio and that of the benchmark, i.e. $S\&P500$ over each investment horizon. If the fitted sharp ratio is positive, we can say that our mean-variance strategy outperforms the benchmark over that investment period.

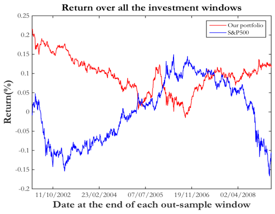


Figure 19
Returns at the end of each investment horizon

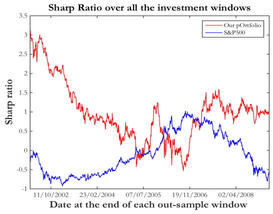


Figure 20
Sharp ratios at the end of each investment horizon

From the two figures above, we can say that during most of time over which we tested, our strategy outperforms $S\&P500$, except the year 2006. Additionally, the outperformance period is relative longer in terms of sharp ratio due to the low variance of our strategy.

If we implement daily dynamic portfolio strategy, i.e, we change the weight vector of our strategy every day, nevertheless the investment horizon still remains 900 days. Thus we can regard the return of strategy at each end of investment horizon as time series and hence evaluate its performance by regressing them of the benchmark to estimate α such that

$$Strategy - r_f = \alpha + \beta(S\&P500 - r_f) \tag{3}$$

where r_f is the risk-free rate and we obtain $\alpha = \mathbf{0.0483}$, which means that our strategy would gain on average 4.83% annually more than $S\&P500$.

8. SUPPORT VECTOR MACHINE

In this section, we try to apply the support vector machines (SVM), which is a powerful machine learning tool introduced during lectures, to help us distinguish *good days* and *bad days* according to our collected COMP dataset.

The COMP datasets include daily prices of ten important commodities such as natural gas, oil and gold, from 2000-01-03 to 2016-11-30. In this section, we firstly tag the data – we consider the day that has more than 4 commodities that have higher prices than corresponding averages as the *good days*, then supplemental days are considered as *bad days*.

Next we split the entire dataset to training dataset and testing dataset, then we implement a SVM via using the python *sklearn* package. We use the Gaussian radial basis function (rbf) kernel as our kernel function and following table summaries our findings: We can observe from the table that when we use 65% of data to train the SVM, we obtain the lowest mistake rate. Hence we believe that when the training data volume is smaller than 65%, the SVM is not well trained, resulting in high mistake rate. However, when the training dataset is too large, the mistake rate is even higher. We believe that this comes from overfitting, i.e., $R^2 \approx 1$, as a result, the mistake rate of the SVM is increased.

Table 4: Testing set size and mistake rate

Testing set size	40%	35%	30%	25%	20%
Mistake Rate	32.5%	20.3%	30.6%	39.7%	65.4%

9. CONCLUSIONS

In this project, we obtained valuable hands-on experiences on processing and studying financial datasets. The size of the financial datasets are typically big hence challenging.

We studied special traits of financial data such as heavy tails and asymmetry of gains and loss. We also implemented a correlation matrix filtering technique backed up by the random matrix theory, which is beneficial in helping us analyzing principle components. We carried out rolling in-sample portfolio optimization and compared the in-sample and out-of-sample risk of portfolios over the same length of data.

In addition, we applied a clustering technique to detect the communities among the stocks and commodities, which is a useful guidance when constructing portfolios. A SVM is implemented to distinguish *good days* and *bad days*. We realized that using too much data to train the model results in overfitting, i.e., $R^2 \approx 1$.

10. ACKNOWLEDGMENTS

The authors would like to thank Prof. Damien Challet, for his patient guidance during the lectures and hand-on sessions. All related codes are available on **here** under the CC license.

11. REFERENCES

[1] Rama Cont. *Empirical properties of asset returns: stylized facts and statistical issues.* Centre de Mathématiques Appliquées, Ecole Polytechnique, F-91128 Palaiseau, France.

[2] Olivier Ledoit and Michael Wolf. *Honey, I Shrunk the Sample Covariance Matrix.*

APPENDIX

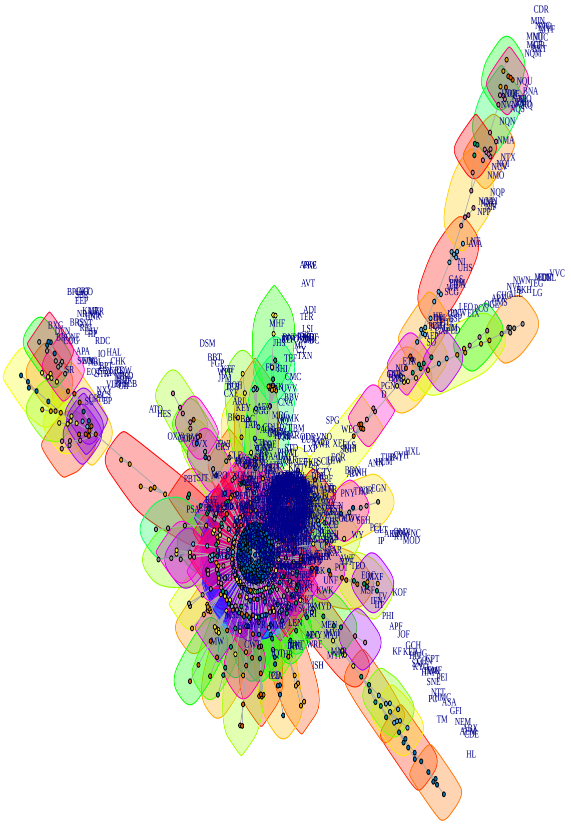


Figure 21: Full communities among US stocks