CSIC 5011: Finding Trend in Stock Market with RobustPCA

Liang Zhicong 20485672
Department of Mathematics, Hong Kong University of Science and Technology

Introduction

Volatility is one of the main character of stock market that people love and hate. Intuitively, the fluctuation of a stock is affected by the general trend of the same class of stocks, and other relatively individual factors. For example, the stock trend of an oil company maybe influenced by the inter-national oil price and its own management. We regard the first factor as the "background" while the second one as "noise".

In this report, we will explore how to capture the main trends within different classes of stocks with RobustPCA. To better evaluate the performance, we convert our problem into a classification task using SNP500 dataset. Specifically, given a stock, we will try to recognize its underlying class (10 classes in total, e.g. Industries, Information Technology, etc.), based on the assumption that stocks in the same class will have similar trend. That is, higher accuracy can be achieved if one could better characterize the trends of stocks.

Through experiment, we find out that RobustPCA can successfully separate the main trend from the noise and achieve the highest accuracy regarding classification, while PCA struggles with the noise and underperforms.

Methods

PCA is one of the most widely used dimension reduction method. Its main aim is to reduce the dimensionality of data while retaining as much as possible of variance in the data.

RobustPCA (RPCA) tries to recover a low-rank component L_0 and sparse component S_0 from their superposition $D = L_0 + S_0$ by solving a convex program called principle component pursuit [1].

We base our experiment on the assumption that stocks will have similar trends within classes while different trends between classes. If the main variance of the data comes from these trends instead of the noise, PCA should be able to capture them. For another, the main trends should be of low rank, since stocks are highly correlated. In this case, RobustPCA can capture this low-rank component, as the background when it applies to video surveillance.

We will perform logistics regression on raw data, features from PCA as well as RobustPCA, and compare the classification result, to evaluate whether they successfully capture the trends.

Dataset

SNP500 consists of a 452-by-1257 matrix with closed price of 452 stocks in workdays for 4 years [2]. These 452 stocks lie in 10 classes: Consumer Discretionary (CD), Consumer Staples(CS), Energy (EN), Financials (FI), Health Care (HE), Industrials (IN), Information Technology (IT), Materials (MA), Telecommunications Services (TE) and Utilities (UT). Their distribution are shown in Figure 1.

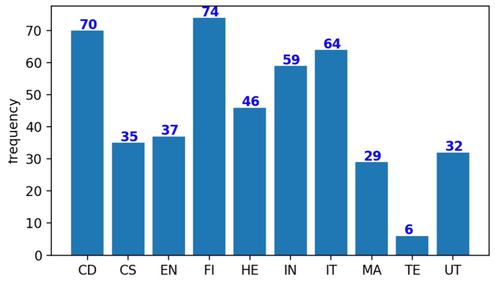

Figure 1. Distribution of stocks regrading classes.

To eliminate the affect of different magnitude among stocks, we will use return as our input, defined as

$$R_t = (X_t - X_{t-1})/X_{t-1}$$

We visualize 10 stocks from two classes in Figure 2. Here, we find that stocks from Information Technology are more volatile than those from Health Care, especially at the beginning, which validates our basic assumption.

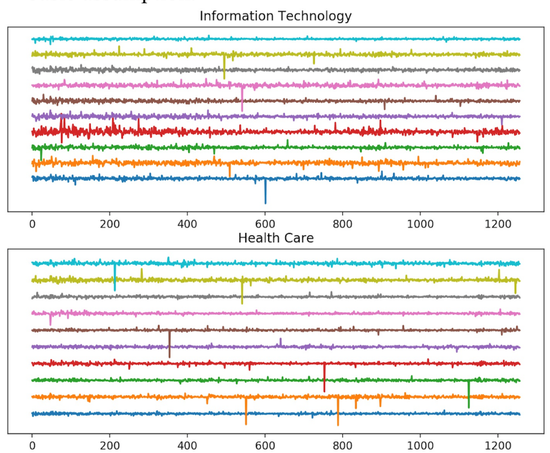

Figure 2. Visualizing 10 stocks from IT and HE respectively. The horizon is zero.

Experiment

Here we perform classification using logistic regression on raw data, features from PCA and features from low-rank component of RobustPCA [4]. We use 300 stocks as our training set while the remaining 152 as our test set. Results are reported under the best hyper-parameter settings with 5 runs of different random seeds, shown in Table 1.

	Train ACC	Test ACC
Raw Data	1.0000 ± 0.000	0.8646 ± 0.005
PCA (n=150)	0.9946 ± 0.002	0.8616 ± 0.006
RPCA ($\lambda = 1257^{-0.5}$)	0.9270 ± 0.000	**0.8750 ± 0.000**
RPCA(by class)*	0.9870 ± 0.000	0.9800 ± 0.000

Table 1. Classification result. RPCA(by-class) means we perform RPCA on the data within each class. Since it already used the class information before classification, its result is not comparable to the others. We show it here for reference.

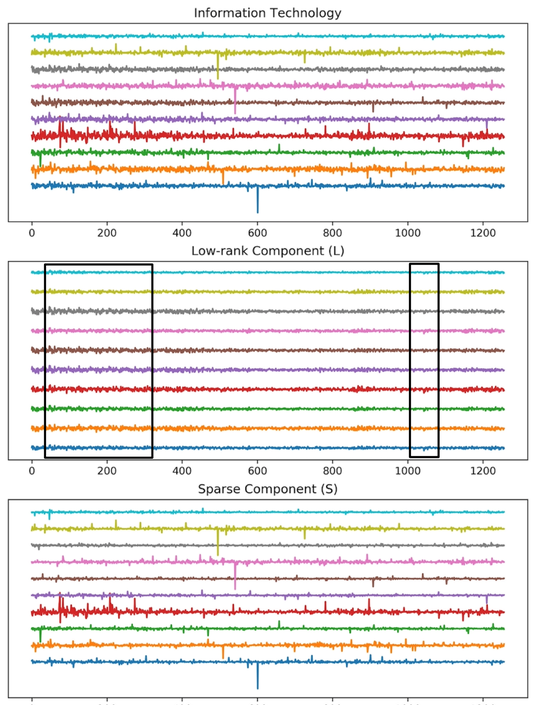

Figure 3. Decomposition of 10 stocks in IT by RPCA. Similar trends are enclosed by black boxes.

Analysis

From Table 1, we find that PCA performs even worse than raw input. We attribute this underperformance to the non-negligible noise of stock market. That is, since PCA tries to capture variance as much as possible, they may be even noise-pursuit, especially when dealing with high-variance data like stock price. And we may refer to this phenomenon as phase transitions of PCA [3].

From Figure 2, we find that the low-rank components (the middle figure) of 10 stocks from IT have similar pattern (enclosed by black boxes) that is hard to notice in raw data. Separating out these main trends, our classifier achieve the best result as shown in Table 1.

Other statistics we want to list here is that, in PCA, 150 components account for 89.5% variance of the data. As for RobustPCA, if we use $\lambda = \max(452, 1257)^{-0.5}$ as the paper [1], we achieve the result in Table 2, while the rank and sparsity of L_0 is 265 and 0.5%, and sparsity of S_0 is 24.9%. However, if we choose $\lambda = 425^{-0.5}$, the classification accuracy is 1.3% higher and L_0 is of rank 452 (the same as raw input) and of sparsity 1.1%, while sparsity of S_0 raises to 83.7%. This indicate the trade-off between rank of L_0 and sparsity of S_0. Even though the latter setting has higher accuracy, we believe that the former one aligns more with our basic assumption that trends are of low rank.

Another founding is that a large proportion (47.4%) of classification error relates to industrial stocks. We attribute this phenomenon to the fact that industries are highly correlated to other fields like materials, IT and consumer goods. In this case, industrial stocks have weaker identity and are easily influenced by other stocks, resulting a higher classification error.

References

[1] Candès, E.J., Li, X., Ma, Y. and Wright, J., 2011. Robust principal component analysis?. Journal of the ACM (JACM), 58(3), p.11.

[2] SNP dataset: https://yao-lab.github.io/data/snp452-data.mat

[3] Yao, Y. A mathematical Introduction of Data Science. Retrieved from https://github.com/yao-lab/yao-lab.github.io/blob/master/book_datasci.pdf

[4] Source Code: https://github.com/ZhicongLiang/SNP500-Stock-Trend