Portfolio Selection via Text Based Network

June 30, 2021

Abstract

Keywords:
JEL Codes:

1 Introduction

The covariance matrix is important in many areas, like constructing a Mean-Variance portfolio. High dimensional covariance matrix Σ_T contains huge estimation error given small data X_T. One potential method to solve this problem is shrinkage method[?]. One simple way to do this would be taking the convex combination between covariance matrix and a shrinkage target. This target could be identity matrix or other customer-tailered matrix. Upon this approach we can reduce the variance of sample covariance matrix because of introducing an biased but low-variant item. An alternative aspect thinking this shrinkage method would be a trade off between bias and variance.

1.1 Shrinkage

One of the reason why shrinkage method would work out is that it impose market structure, to some extent, into the covariance matrix estimation. The sample covariance matrix is approximately an Maximum Likelihood Estimator(MLE) of historical data without any structure. It means this estimator does not exploit any market features. It would lead to poor estimation under small data situation. Shrinkage methods outperform sample covariance matrix by exploiting market information.

Ledoit and Wolf[?] solve two important problems about shrinkage method. They are what kind of features to be included and how much should be imposed. They tried with identity matrix or single-index model as shrinkage target to decide which information to be included into the estimation. And a shrinkage intensity is decided to control how much of features should be imposed.

1.2 Text Based Network

However, the shrinkage targets Ledoit and Wolf used are limited, as they imposed limited information into the estimation. A better designed shrinkage target would lead to better performance. So we propose using Text-Based Network(TBN) as shrinkage target. Proposed by Hoberg and Phillips[?] TBN describes industries boundaries by using a square correlation matrix. They create this correlation matrix by parsing 10-K business description report that each firm submits to SEC annually and name this matrix as Text-Based Network. By assuming firm product is at the core when clustering industries, 10-K report is the public resources suitable to used for drawing industries boundaries. And in the correlation matrix, each value indicates the product similarity of a pair of firms.

Using TBN as shrinkage target would improve the shrinkage estimator in both estimation error and adding new information. First, TBN is generated from parsing 10-K report which updates annually. So this correlation matrix doesn't change within a year. Also this correlation doesn't change dramatically between each year while their stock returns have more volatility. This low volatility would hence reduce the shrinkage estimation error. Second, TBN utilizes text data and add new information to the estimator making it further to the true covariance matrix. In previous shrinkage estimator proposed by Ledoit and Wolf[?], most are generate within the same data set, which is stock returns, as sample covariance matrix. Hence, the contribution of these shrinkage targets are limited because of limited information sources.

1.3 Minimum Spanning Tree

1.4 Agent Based Learning

2 Methodology

2.1 Shrinkage

Let $\mathbf{H}_t$ denotes covariance matrix that used to construct GMV portfolio.[**?**][**?**]

$$\mathbf{H}_t = \mathbf{D}_t \mathbf{R}_t \mathbf{D}_t \tag{1}$$

where $\mathbf{D}_t$ denoting the diagonal matrix of volatilities and $\mathbf{R}_t$ is the correlation matrix. By shrinking the correlation matrix $\mathbf{R_t}$,

$$\tilde{\mathbf{R}}_t = (1-\alpha)\mathbf{R}_t + \alpha \mathring{\mathbf{R}}_t \tag{2}$$

We can shrink the covariance matrix $\mathbf{H_t}$ indirectly,

$$\tilde{\mathbf{H}}_t = \mathbf{D}_t \tilde{\mathbf{R}}_t \mathbf{D}_t = \mathbf{D}_t \left[(1-\alpha)\mathbf{R}_t + \alpha \mathring{\mathbf{R}}_t\right] \mathbf{D}_t = (1-\alpha)\mathbf{H}_t + \alpha \mathbf{D}_t \mathring{\mathbf{R}}_t \mathbf{D}_t \tag{3}$$

We can use shrank covariance matrix $\tilde{\mathbf{H}}_t$ to construct new shrank GMV portfolio.

$$\begin{aligned} \mathbf{x}_t(\alpha) &= \frac{\tilde{\mathbf{H}}_t^{-1}\mathbf{1}}{\mathbf{1}'\tilde{\mathbf{H}}_t^{-1}\mathbf{1}} \\ &= \frac{\mathbf{D}_t^{-1}\left[(1-\alpha)\mathbf{R}_t + \alpha \mathring{\mathbf{R}}_t\right]^{-1}\mathbf{D}_t^{-1}\mathbf{1}}{\mathbf{1}'\mathbf{D}_t^{-1}\left[(1-\alpha)\mathbf{R}_t + \alpha \mathring{\mathbf{R}}_t\right]^{-1}\mathbf{D}_t^{-1}\mathbf{1}} \end{aligned}$$

As shown above, $\mathbf{x}_t(\alpha)$ is a function of α, $\mathbf{R}_t$ and $\mathring{\mathbf{R}}_t$. Here, $\mathbf{R}_t$ and $\mathring{\mathbf{R}}_t$ is given from data and α is decision variable that we want to control to optimize our objective.

2.2 Minimum Spanning Tree

One fundamental drawback of constructing correlation matrix from news data is that it involves more noise. This noise comes from several aspect, e.g. news accuracy, source, bias and etc. To remove the noise from this news-base network, we can just keep the most significant relationship for each ticker and kick out the rest. An ideal approach to solve this problem, which comes from graph theory, is Minimum Spanning Tree(MST).

Each ticker can be seen as an node in the graph, and the correlation value of one pair is the weight on the edge linking two nodes. MST keep only one edge, which has minimal cost, between two nodes. In this way, 30 tickers are linked by 29 non-zero weights(correlation). Thus keep the most important relationship and removing the noise.

To construct a minimum spanning tree from stocks' return, there're two steps. First, a distance matrix is needed. And it can be converted from correlation matrix.

$$d^z(i,j) = \sqrt{1 - |\rho_{ij}^z|}$$

Where ρ is correlation value for stock pair (i,j) and $d^z(i,j)$ is the converted distance under z norm. Then a minimum spanning tree(MST) can be performed on this distance matrix to select which edge to be kept. We use Kruskal algorithm to construct this MST, and it's an edge-centre algorithm.

1. Initialise the tree as a disconnected graph made up of the nodes in the distance graph
2. Sort edges of the distance graph in ascending order and place in a list
3. For each edge between i and j in this sorted list
   (a) If i and j are not in the same component in the tree, add this edge into the tree

2.3 Reinforcement Learning

Reinforcement Learning(RL) can be used for this non-linear optimization problem finding the optimal α with changing time period. In particular, two kinds of RL method could be applied.One is Value-based methods like Q-learning, the other one is Policy-based methods likepolicy gradient. While leaving the discussion between these two kinds of methods, we firstly introduce the basic elements for RL problem.

- State space $S = \{r_{p,t}\}$
- Action space $A = [0,1]$
- Reward $R_t = \frac{\mathbf{E}[r_p - r_f]}{\sigma[r_p - r_f]}$
- Objective function $\max_{\pi} E_{\pi}\left[R_1 + \gamma R_2 + \cdots + \gamma^{T-1} R_T\right]$

Where, $\{r_{p,t}\}$ is portfolio return. π is policy, which is discussed under each method below. More measurable variables could be added into state space to attain a more precise description of exposure to information environment. Also, reward at each time R_t could be log return or some risk-adjusted measurement.

The first method, Value-based methods like Q-learning[?], can be firstly applied for gaining some insight toward this problem while searching α^* in a discrete space. Taking SARSA[?] as an example, for each episode we go through the training data once. For each step in an episode the agent observes current state S_t and take action A_t according to current policy π. Then agent observes reward R_{t+1} and next time state S_{t+1}, and perform another action A_{t+1} again. After gather these information we are ready to update our Q function, the value for each state and action pair. The updating rule described as below.

$$Q\left(S_t, A_t\right) \leftarrow Q\left(S_t, A_t\right) + \alpha\left[R_{t+1} + \gamma Q\left(S_{t+1}, A_{t+1}\right) - Q\left(S_t, A_t\right)\right]$$

Where α is learning rate. We can see this as an approximation for value function

$$Q(s, a) = \mathbb{E}\left[R_{t+1} + \gamma Q\left(S_{t+1}, A_{t+1}\right) \mid S_t = s, A_t = a\right]$$

And our policy for value-based method is deterministic all the time.

$$\pi(a \mid s) = \begin{cases} \epsilon/m + 1 - \epsilon & \text{if } a = \underset{a' \in \mathcal{A}}{\operatorname{argmax}} Q(s, a') \\ \epsilon/m & \text{otherwise} \end{cases}$$

where ϵ is a predefined probability to choose an action at random. m is the number of action could be choosen randomly.

In summary, we obtain a huge table of Q value for each state and action pair (s, a). Each iteration we going through training data, we update this Q table until it doesn't change any more. After the convergence, our desired policy π^* is straight forward to simply act greedily to choose the action A_t giving us highest Q value $Q(S_t, A_t)$ under each state S_t.

After gaining some insight from using value-based methods, policy-based methods like policy gradient can be used. And policy gradient method would perform better, since it works better on continuous space. Policy gradient method share a similar structure while don't bother with value function $Q(s, a)$ or $V(s)$. Instead we introduce a new variable θ named policy parameter, which governs the new policy function $\pi(a \mid s, \theta)$. A new performance function $J(\theta)$ is also introduced to measure agent's performance. We can choose our performance function as

$$J(\theta) = E_{\pi_\theta}\left[R_1 + \gamma R_2 + \cdots + \gamma^{T-1} R_T\right]$$

For implementing policy gradient we take REINFORCE algorithm as an example. In each episode we go through the training data once. For each step in an episode, the agent takes action A_t given current state S_t and policy $\pi(a \mid s, \theta)$, then observes reward R_{t+1} and next state S_{t+1}. This process is looped til the end of each episode. At the end of each episode, the agent update its policy by updating policy parameter θ_t

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J\left(\theta_t\right)}$$

where α is learning rate. We use $\widehat{\nabla J(\theta_t)}$the gradient of performance function with respect to policy parameter θ_t to update θ_t. We learn the value of θ which controls the policy giving us the optimal policy.

3 Empirical Results

3.1 Data

3.2 Minimum Spanning Tree

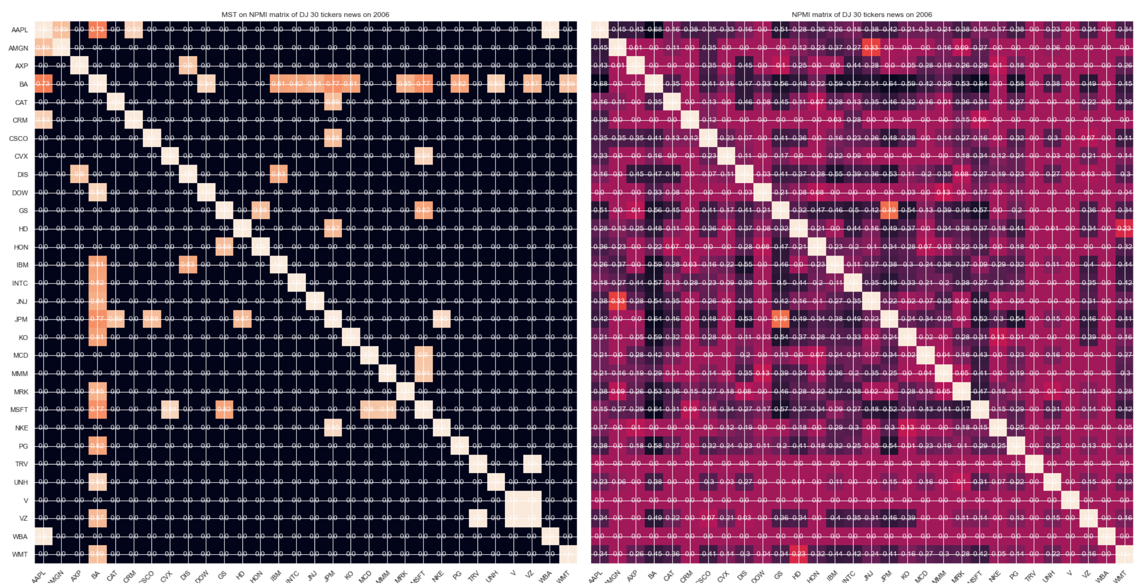


Figure 1: Comparison of MST and NPMI matrix on 2006

As shown in figure1, after performing MST NPMI matrix becomes an sparse matrix. Only strong relationship are kept.

A subset of MST for NPMI matrix sample is shown below in table **??**

	AAPL	AMGN	AXP	BA	CAT	CRM	CSCO	CVX	DIS	DOW
AAPL	1.000	0.892	0.000	0.730	0.0	0.926	0.0	0.0	0.000	0.000
AMGN	0.892	1.000	0.000	0.000	0.0	0.000	0.0	0.0	0.000	0.000
AXP	0.000	0.000	1.000	0.000	0.0	0.000	0.0	0.0	0.896	0.000
BA	0.730	0.000	0.000	1.000	0.0	0.000	0.0	0.0	0.000	0.944
CAT	0.000	0.000	0.000	0.000	1.0	0.000	0.0	0.0	0.000	0.000
CRM	0.926	0.000	0.000	0.000	0.0	1.000	0.0	0.0	0.000	0.000
CSCO	0.000	0.000	0.000	0.000	0.0	0.000	1.0	0.0	0.000	0.000
CVX	0.000	0.000	0.000	0.000	0.0	0.000	0.0	1.0	0.000	0.000
DIS	0.000	0.000	0.896	0.000	0.0	0.000	0.0	0.0	1.000	0.000
DOW	0.000	0.000	0.000	0.944	0.0	0.000	0.0	0.0	0.000	1.000

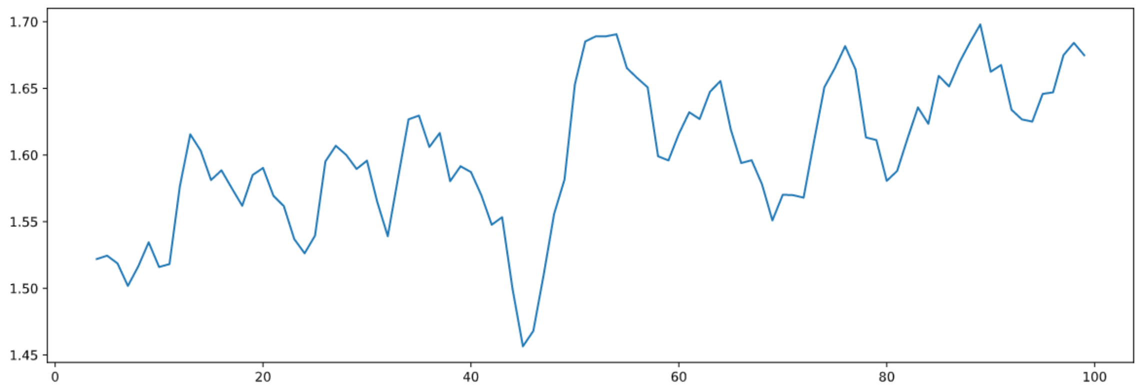
Table 1: MST from NPMI matrix(subset)

3.3 RL Control

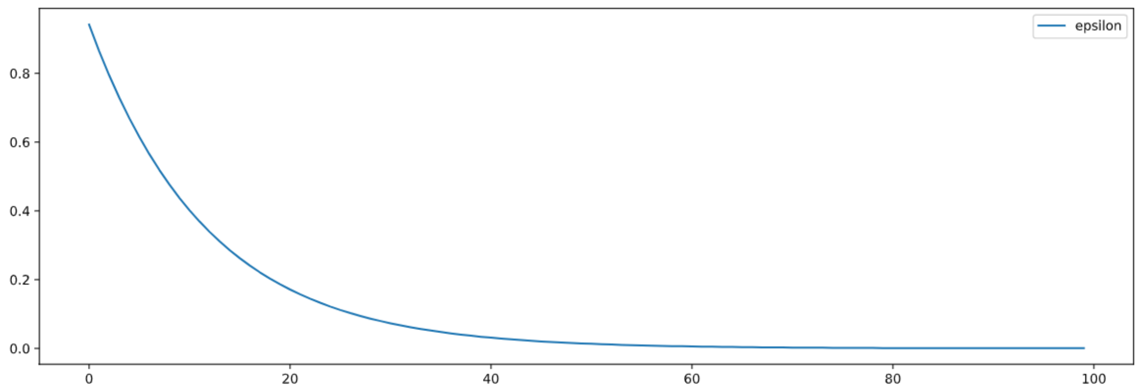
In our Reinforcement Learning prediction experiment, we use two kinds of methods to predict optimal shrinkage intensity on each year. The two specific algorithms are Deep Q Network(DQN) and REINFORCE(pg). In following subsections, we firstly show the training results then test the trained agents on historical data to predict optimal shrinkage intensity for each year. Finally we provide our analysis based on previous results.

3.3.1 Training results

We first show the training figures for DQN model. In the plot below, while the total rewards for each episode is very noisy, it has an upward trend. It seems that the agent learns something from experience.



(a) Total Rewards for Each episode



(b) Decaying Epsilon(exploring rate) for Each episode

Figure 2: Training figure of DQN agent from 1996 to 2013

We then implement the REINFORCE(pg) model, and the training figure is shown below. It has a similar shape with DQN agent. Both are very noisy in training performance. And they don't have sign of converge. We would leave the discussion of the reason to the next section.

3.3.2 Test Performance

We test both trained agents on historical data to predict optimal shrinkage intensity for each year. And compare their actions to the true value, which obtained from in-sample analysis. And we use the 'true value' serves as the benchmark.

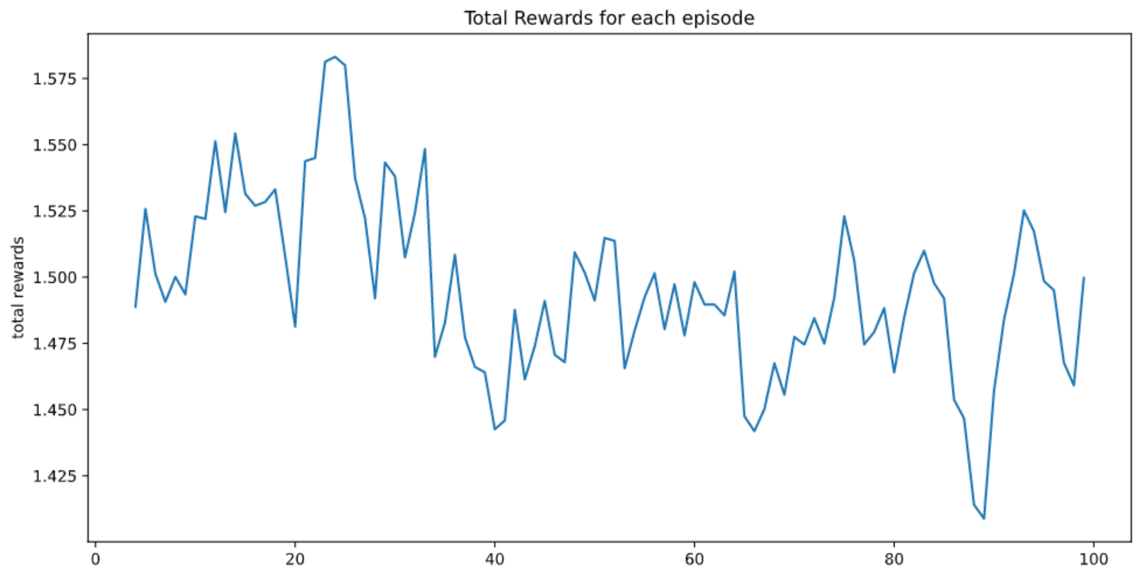
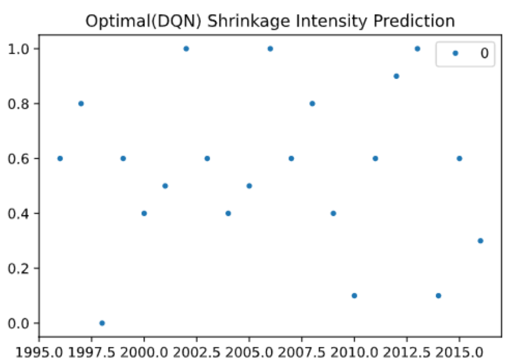
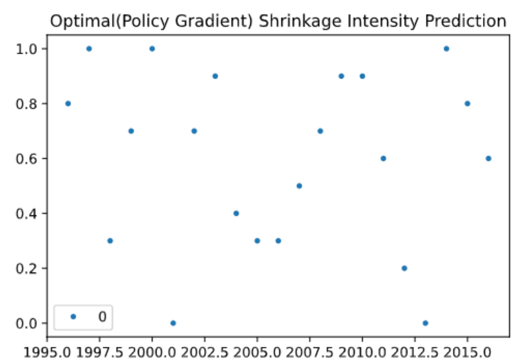


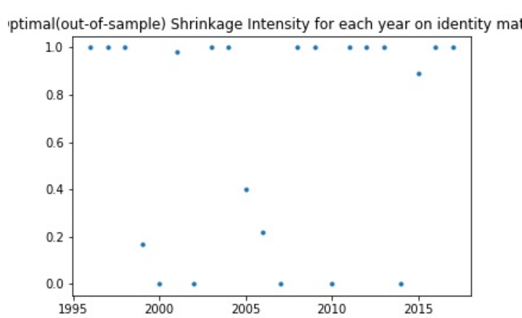
Figure 3: Training figure of PG agent from 1996 to 2013



(a) DQN prediction on shrinkage intensity



(b) PG prediction on shrinkage intensity



(c) True optimal shrinkage intensity

Figure 4: comparison between RL agents performance

3.4 Back-testing

So far we perform three technique to improve from plain frequency matrix and obtain three new representation of correlation matrix, Normalized point-wise Information(NPMI), Minimum Spanning Tree(MST) and Kalman filter. We need to back test these correlation matrix to construct portfolios and compare their performance on the real market. First we need to construct portfolio from correlation matrix.

$$\tilde{\mathbf{H}}_t = \mathbf{D}_t \tilde{\mathbf{R}}_t \mathbf{D}_t$$
$$\mathbf{x}_t(\alpha) = \frac{\tilde{\mathbf{H}}_t^{-1}\mathbf{1}}{\mathbf{1}'\tilde{\mathbf{H}}_t^{-1}\mathbf{1}}$$

Where,

- $\mathbf{D}_t$ is the diagonal matrix of volatilities
- $\tilde{\mathbf{R}}_t$ is correlation matrix
- $\mathbf{H}_t$ is covariance matrix
- $\mathbf{x}(\alpha)$ is global mean variance(GMV) Portfolio

Then we need to test these portfolios on the next year stock market to get the portfolios' return. Here we use an risk adjusted performance, Sharp ratio, as the performance evaluation for each portfolio.

Following above we can construct GMVP $\mathbf{x}_t(\alpha)$ at the end of each year for a givce shrinkage intensity α. Hence we can compute out-of-sample GMVP return $r_{p,t+1}$ using following year's stock returns $\mathbf{Y}_{t+1}$. The concrete equation is described bellow,

$$r_{p,t+1}(\alpha) = \mathbf{Y}_{t+1}\mathbf{x}_t(\alpha) \tag{4}$$

Further, we can compute Sharpe ratio $SR(\alpha)$ for different time series $r_{p,t+1}$ given different α.

$$SR = \frac{\mathbf{E}[r_p - r_f]}{\sigma[r_p - r_f]} \tag{5}$$

Given a concrete time series r_p, we output a Sharpe ratio as above. Here, the risk free rate is provided by Kenneth R. French - Data Library. We repeat this process to back test on selected time range.

It's interesting that original stock correlation matrix, which serves as an benchmark, is not too bad. The portfolio constructed from frequency matrix, which serves as secondary benchmark, has a similar performance with first benchmark.

One good thing is that portfolio construct from NPMI achieves the highest score and way better than any other one. And we think the consideration of joint and marginal probability contribute most for this huge jump. What astonishes us is minimum spanning tree(MST) approach lag way behind the benchmark. It seems that we not only removing the noise but also many useful information.

What's more, Kalman filter method performs not bad. Although it has similar score with benchmark, but it beat heavily the two original data source approach(News MST and stock MST). And we can expect much better performance from Kalman filter, if it combine two better information(News NPMI and Stock correlation).

We further backtest on MST, at the beginning of each year a distance matrix and a corresponding MST are constructed from last year correlation matrix. The MST is used as the new correlation matrix to construct GMVP. We backtest from 1996 to 2016, and the Sharpe ratio for the whole period are reported in below table for performance analysis.

Table 2: Sharpe ratio on DOW 30 from 1997 to 2018

Naive Shrinkage		
Strategy	TBN	Identity
shrink 0 pct	0.444084	0.444084
shrink 50 pct	0.349595	0.507769
shrink 100 pct	-0.43066	0.573233
Ledoit &Wolf		
linear shrinkage	NA	0.471745
non-linear	NA	0.449374
Minimum Spanning Tree		
MST on stock		MST on TBN
-0.056445		0.667070
Reinforcement Learning		
DQN	0.122987	0.506485
REINFORCE	-0.490042	0.503571

3.5 Portfolio turnover

Portfolio turnover measures the frequency of a portfolio changing its position. In the paper, we re-balance our portfolio annually. So it measures the magnitude of changing positions for each year. The concrete formula is shown below.

$$\hat{W} = \sum_{i=1}^{N} (w_{i,t+1} - w_{i,t}) 1_{\{w>0\}} \tag{6}$$

$$\tilde{W} = \sum_{i=1}^{N} (w_{i,t+1} - w_{i,t}) 1_{\{w<0\}} \tag{7}$$

$$W = min\{\hat{W}, \tilde{W}\} \tag{8}$$

$$P = \frac{P_{t+1} + P_t}{2} \tag{9}$$

$$TO = \frac{W}{P} \tag{10}$$

Where $w_{i,t}$ is the value for holding stock i at time t. And P_t is the portfolio value at time t. So the turnover is the minimum of securities bought or sold divided by portfolio average value. We calculate the turnover on different strategies and the backtesting period is from 1998 to 2018. The turnover is calculated annually and an average is reported in Table 3.

Table 3: Turn over of portfolio from 1998 to 2018

Naive Shrinkage		
Strategy	TBN	Identity
shrink 0 pct	0.803780	0.803780
shrink 50 pct	0.391381	0.064996
shrink 100 pct	0.158527	0.000000
Ledoit &Wolf		
linear shrinkage	0.726502	0.684099
non-linear	NA	0.679570
Minimum Spanning Tree		
MST on stock		MST on TBN
4.672345		3.451522
Reinforcement Learning		
DQN	NA	NA
REINFORCE	NA	NA

When shrink half way or fully to the TBN, the portfolio turnover is very small, which is below 0.5. That means these portfolio don't change their position frequently and largely. It's because of the stability of TBN, which doesn't change much over years. While Ledoit Wolf's linear shrinkage strategy results in a much higher turnover rate, because the estimated shrinkage intensity is lower than 50 percentage on the backtesting period.

However, the turnover for MST strategies are much higher than others, which is suspicious. A double check is needed here.

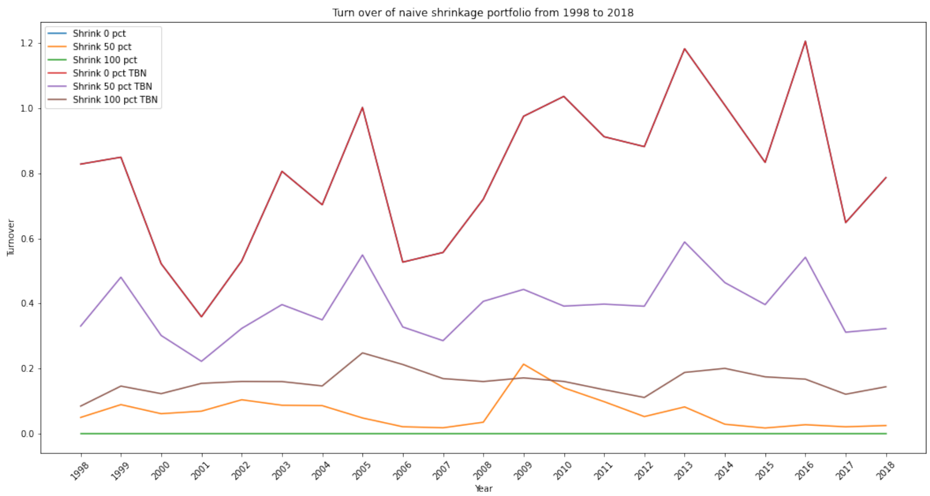


Figure 5: Turn over of naive shrinkage portfolio from 1998 to 2018

Time series of above strategies are reported in Figure 5, 6 and 7. For naive shrinkage strategies, the more shrink to the shrinkage target the lower and more stable of turnover over the time. For Ledoit and Wolf's shrinkage methods, all strategies have similar turnover over the time. That is due to the similar shrinkage intensity estimation.

However, the MST strategies' time series has a suspicious behaviour, which is in line with the result in table. A double check is needed here.

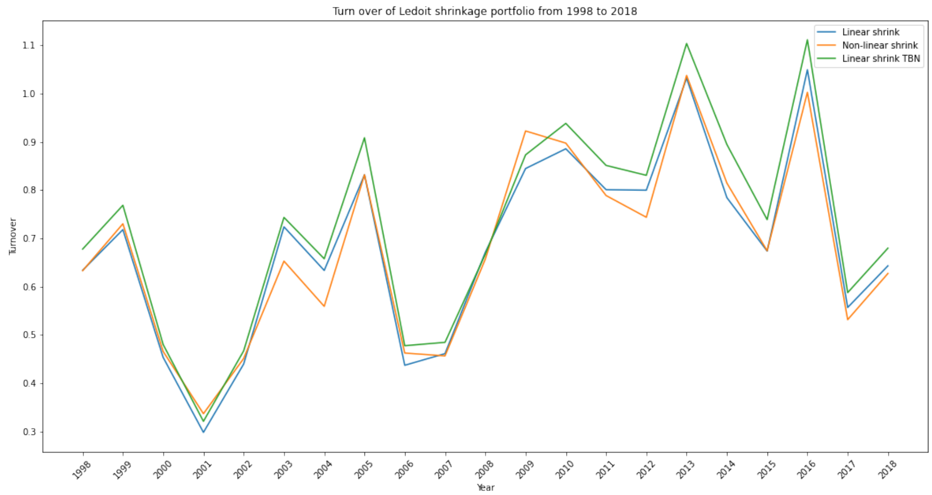


Figure 6: Turn over of Ledoit shrinkage portfolio from 1998 to 2018

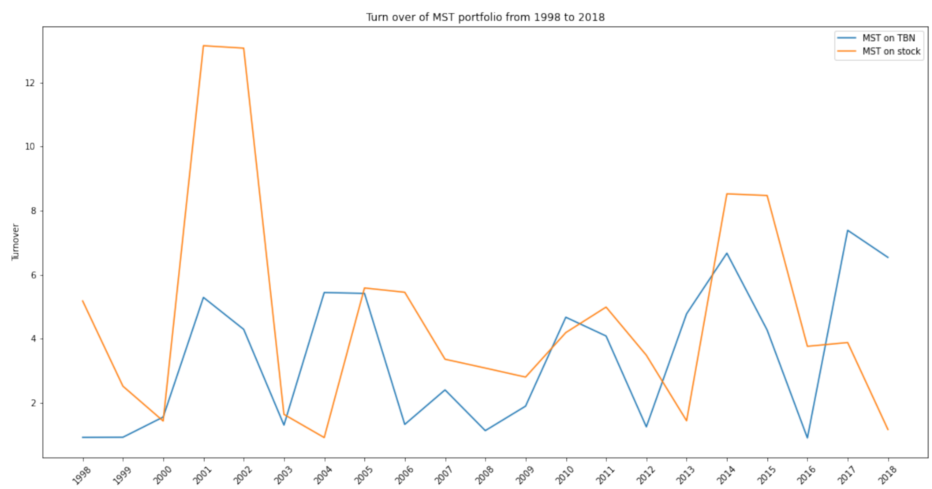


Figure 7: Turn over of MST portfolio from 1998 to 2018

4 Conclusion

Appendix

A comparison between MST and original correlation matrix is shown below.

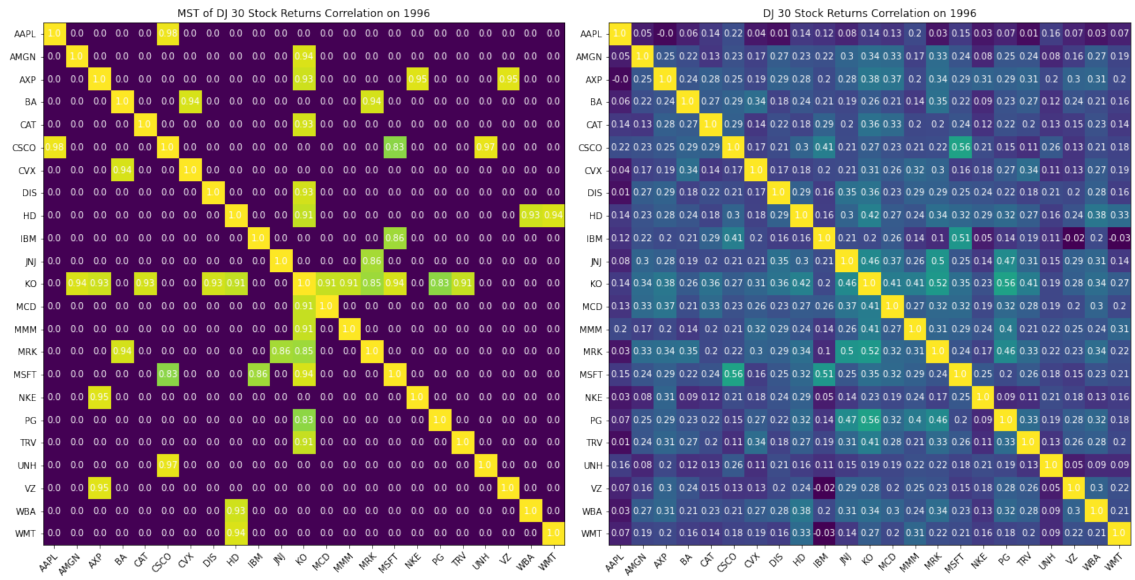


Figure 8: MST and correlation matrix for stock return on 1996

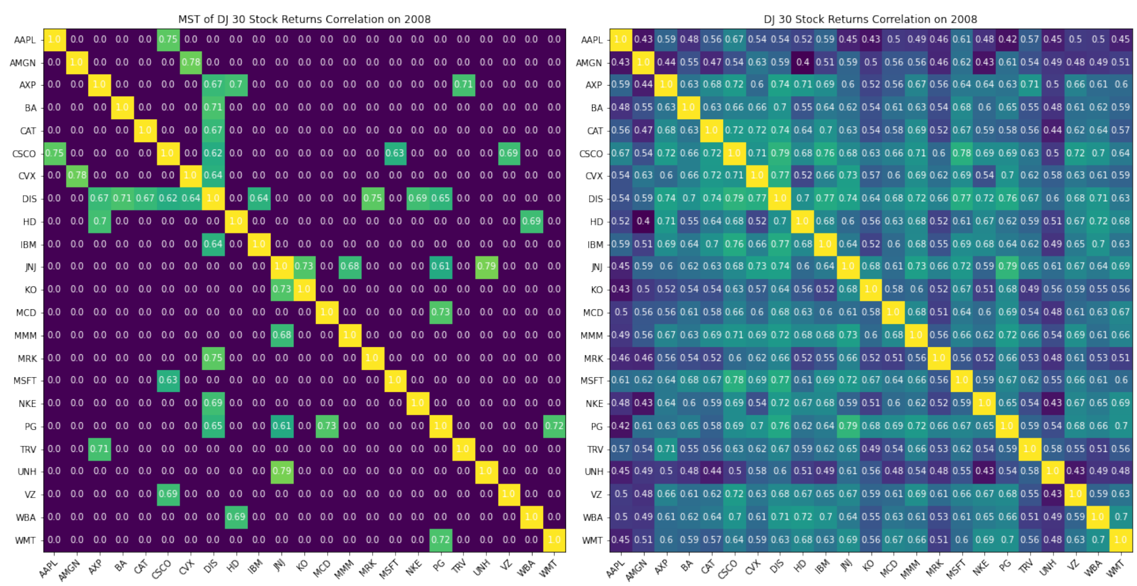


Figure 9: MST and correlation matrix for stock return on 2008

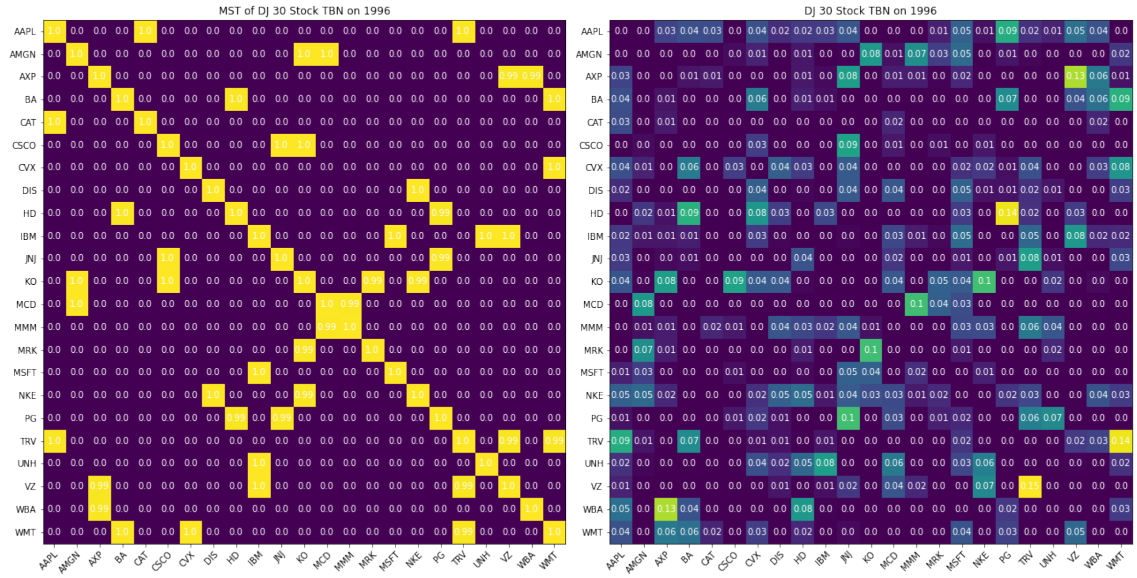


Figure 10: MST and correlation matrix for TBN on 1996

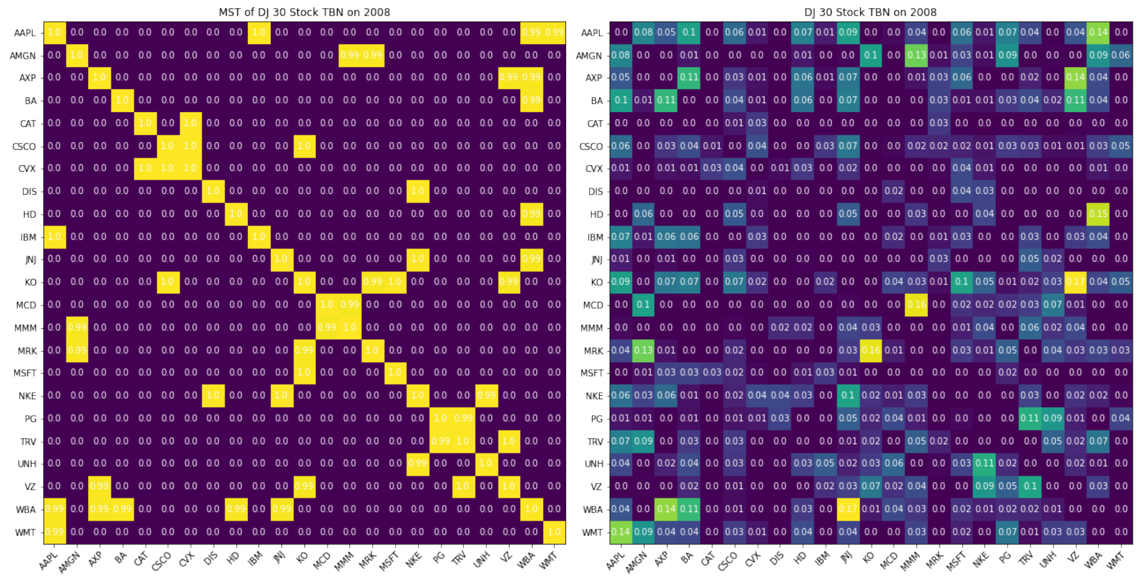


Figure 11: MST and correlation matrix for TBN on 2008

Table 4: correlation matrix of stock return in 1996

	AAPL	AMGN	AXP	BA	CAT	CSCO	CVX	DIS	HD	IBM	INTC	JNJ	JPM	KO	MCD	MMM	MRK	MSFT	NKE	PG	TRV	UNH	VZ	WBA	WMT
AAPL	1.00	0.05	-0.00	0.06	0.14	0.22	0.04	0.01	0.14	0.12	0.24	0.08	0.08	0.14	0.13	0.20	0.03	0.15	0.03	0.07	0.01	0.16	0.07	0.03	0.07
AMGN	0.05	1.00	0.25	0.22	0.13	0.23	0.17	0.27	0.23	0.22	0.09	0.30	0.32	0.34	0.33	0.17	0.33	0.24	0.08	0.25	0.24	0.08	0.16	0.27	0.19
AXP	-0.00	0.25	1.00	0.24	0.28	0.25	0.19	0.29	0.28	0.20	0.13	0.28	0.44	0.38	0.37	0.20	0.34	0.29	0.31	0.29	0.31	0.20	0.30	0.31	0.20
BA	0.06	0.22	0.24	1.00	0.27	0.29	0.34	0.18	0.24	0.21	0.17	0.19	0.30	0.26	0.21	0.14	0.35	0.22	0.09	0.23	0.27	0.12	0.24	0.21	0.16
CAT	0.14	0.13	0.28	0.27	1.00	0.29	0.14	0.22	0.18	0.29	0.19	0.20	0.17	0.36	0.33	0.20	0.20	0.24	0.12	0.22	0.20	0.13	0.15	0.23	0.14
CSCO	0.22	0.23	0.25	0.29	0.29	1.00	0.17	0.21	0.30	0.41	0.50	0.21	0.29	0.27	0.23	0.21	0.22	0.56	0.21	0.15	0.11	0.26	0.13	0.21	0.18
CVX	0.04	0.17	0.19	0.34	0.14	0.17	1.00	0.17	0.18	0.20	0.08	0.21	0.27	0.31	0.26	0.32	0.30	0.16	0.18	0.27	0.34	0.11	0.13	0.27	0.19
DIS	0.01	0.27	0.29	0.18	0.22	0.21	0.17	1.00	0.29	0.16	0.18	0.35	0.31	0.36	0.23	0.29	0.29	0.25	0.24	0.22	0.18	0.21	0.20	0.28	0.16
HD	0.14	0.23	0.28	0.24	0.18	0.30	0.18	0.29	1.00	0.16	0.20	0.30	0.30	0.42	0.27	0.24	0.34	0.32	0.29	0.32	0.27	0.16	0.24	0.38	0.33
IBM	0.12	0.22	0.20	0.21	0.29	0.41	0.20	0.16	0.16	1.00	0.44	0.21	0.26	0.20	0.26	0.14	0.10	0.51	0.05	0.14	0.19	0.11	-0.02	0.20	-0.03
INTC	0.24	0.09	0.13	0.17	0.19	0.50	0.08	0.18	0.20	0.44	1.00	0.19	0.17	0.19	0.12	0.23	0.11	0.60	0.19	0.10	0.18	0.22	0.09	0.07	0.21
JNJ	0.08	0.30	0.28	0.19	0.20	0.21	0.21	0.35	0.30	0.21	0.19	1.00	0.45	0.46	0.37	0.26	0.50	0.25	0.14	0.47	0.31	0.15	0.29	0.31	0.14
JPM	0.08	0.32	0.44	0.30	0.17	0.29	0.27	0.31	0.30	0.26	0.17	0.45	1.00	0.50	0.40	0.26	0.45	0.34	0.18	0.36	0.33	0.20	0.30	0.32	0.23
KO	0.14	0.34	0.38	0.26	0.36	0.27	0.31	0.36	0.42	0.20	0.19	0.46	0.50	1.00	0.41	0.41	0.52	0.35	0.23	0.56	0.41	0.19	0.28	0.34	0.27
MCD	0.13	0.33	0.37	0.21	0.33	0.23	0.26	0.23	0.27	0.26	0.12	0.37	0.40	0.41	1.00	0.27	0.32	0.32	0.19	0.32	0.28	0.19	0.20	0.30	0.20
MMM	0.20	0.17	0.20	0.14	0.20	0.21	0.32	0.29	0.24	0.14	0.23	0.26	0.26	0.41	0.27	1.00	0.31	0.29	0.24	0.40	0.21	0.22	0.25	0.24	0.31
MRK	0.03	0.33	0.34	0.35	0.20	0.22	0.30	0.29	0.34	0.10	0.11	0.50	0.45	0.52	0.32	0.31	1.00	0.24	0.17	0.46	0.33	0.22	0.23	0.34	0.22
MSFT	0.15	0.24	0.29	0.22	0.24	0.56	0.16	0.25	0.32	0.51	0.60	0.25	0.34	0.35	0.32	0.29	0.24	1.00	0.25	0.20	0.26	0.18	0.15	0.23	0.21
NKE	0.03	0.08	0.31	0.09	0.12	0.21	0.18	0.24	0.29	0.05	0.19	0.14	0.18	0.23	0.19	0.24	0.17	0.25	1.00	0.09	0.11	0.21	0.18	0.13	0.16
PG	0.07	0.25	0.29	0.23	0.22	0.15	0.27	0.22	0.32	0.14	0.10	0.47	0.36	0.56	0.32	0.40	0.46	0.20	0.09	1.00	0.33	0.19	0.28	0.32	0.18
TRV	0.01	0.24	0.31	0.27	0.20	0.11	0.34	0.18	0.27	0.19	0.18	0.31	0.33	0.41	0.28	0.21	0.33	0.26	0.11	0.33	1.00	0.13	0.26	0.28	0.20
UNH	0.16	0.08	0.20	0.12	0.13	0.26	0.11	0.21	0.16	0.11	0.22	0.15	0.20	0.19	0.19	0.22	0.22	0.18	0.21	0.19	0.13	1.00	0.05	0.09	0.09
VZ	0.07	0.16	0.30	0.24	0.15	0.13	0.13	0.20	0.24	-0.02	0.09	0.29	0.30	0.28	0.20	0.25	0.23	0.15	0.18	0.28	0.26	0.05	1.00	0.30	0.22
WBA	0.03	0.27	0.31	0.21	0.23	0.21	0.27	0.28	0.38	0.20	0.07	0.31	0.32	0.34	0.30	0.24	0.34	0.23	0.13	0.32	0.28	0.09	0.30	1.00	0.21
WMT	0.07	0.19	0.20	0.16	0.14	0.18	0.19	0.16	0.33	-0.03	0.21	0.14	0.23	0.27	0.20	0.31	0.22	0.21	0.16	0.18	0.20	0.09	0.22	0.21	1.00

Table 5: text based network in 1996

	AAPL	AMGN	AXP	BA	CAT	CSCO	CVX	DIS	HD	IBM	INTC	JNJ	JPM	KO	MCD	MMM	MRK	MSFT	NKE	PG	TRV	UNH	VZ	WBA	WMT
AAPL	1.00	0.00	0.03	0.04	0.03	0.00	0.04	0.02	0.02	0.03	0.01	0.04	0.04	0.00	0.00	0.00	0.01	0.05	0.01	0.09	0.02	0.01	0.05	0.04	0.00
AMGN	0.00	1.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.02	0.02	0.00	0.08	0.01	0.07	0.03	0.05	0.00	0.01	0.00	0.00	0.00	0.00	0.02
AXP	0.03	0.00	1.00	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.03	0.00	0.08	0.00	0.01	0.01	0.00	0.02	0.00	0.00	0.00	0.00	0.13	0.06	0.01
BA	0.04	0.00	0.01	1.00	0.00	0.00	0.06	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.04	0.06	0.09
CAT	0.03	0.00	0.01	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00
CSCO	0.00	0.00	0.00	0.00	0.00	1.00	0.03	0.00	0.00	0.00	0.01	0.00	0.09	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
CVX	0.04	0.01	0.00	0.06	0.00	0.03	1.00	0.04	0.03	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.02	0.02	0.01	0.04	0.00	0.00	0.03	0.08
DIS	0.02	0.00	0.00	0.00	0.00	0.00	0.04	1.00	0.00	0.00	0.00	0.00	0.04	0.00	0.04	0.00	0.00	0.05	0.01	0.01	0.02	0.01	0.00	0.00	0.03
HD	0.00	0.02	0.01	0.09	0.00	0.00	0.08	0.03	1.00	0.03	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.14	0.02	0.00	0.03	0.00	0.00
IBM	0.02	0.01	0.01	0.01	0.00	0.00	0.03	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.03	0.01	0.00	0.05	0.00	0.00	0.05	0.00	0.08	0.02	0.02
INTC	0.03	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.04	0.00	1.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.01	0.08	0.01	0.00	0.00	0.03
JNJ	0.01	0.02	0.03	0.00	0.12	0.01	0.00	0.00	0.00	0.00	0.00	1.00	0.06	0.01	0.00	0.00	0.09	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00
JPM	0.04	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	1.00	0.00	0.00	0.01	0.02	0.05	0.00	0.02	0.00	0.01	0.13	0.07	0.02
KO	0.04	0.00	0.08	0.00	0.00	0.09	0.04	0.04	0.00	0.00	0.06	0.02	0.00	1.00	0.04	0.00	0.05	0.04	0.10	0.00	0.00	0.02	0.00	0.00	0.00
MCD	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	1.00	0.10	0.04	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MMM	0.00	0.01	0.01	0.00	0.02	0.01	0.00	0.04	0.03	0.02	0.00	0.00	0.04	0.01	0.00	1.00	0.00	0.03	0.03	0.00	0.06	0.04	0.00	0.00	0.00
MRK	0.00	0.07	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.10	0.00	0.00	1.00	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.00
MSFT	0.01	0.03	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.09	0.02	0.05	0.04	0.00	0.02	0.00	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
NKE	0.05	0.05	0.02	0.00	0.00	0.00	0.02	0.05	0.05	0.01	0.01	0.05	0.04	0.03	0.03	0.01	0.02	0.00	1.00	0.02	0.03	0.00	0.00	0.04	0.03
PG	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.00	0.00	0.10	0.00	0.03	0.00	0.01	0.02	0.00	1.00	0.06	0.07	0.00	0.00	0.00
TRV	0.09	0.01	0.00	0.07	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	1.00	0.00	0.02	0.03	0.14
UNH	0.02	0.00	0.00	0.00	0.00	0.00	0.04	0.02	0.05	0.08	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.03	0.06	0.00	0.00	1.00	0.00	0.00	0.02
VZ	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.01	0.02	0.00	0.04	0.02	0.00	0.00	0.07	0.00	0.15	0.00	1.00	0.00	0.00
WBA	0.05	0.00	0.13	0.04	0.00	0.00	0.00	0.00	0.08	0.00	0.01	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	1.00	0.03
WMT	0.04	0.00	0.06	0.06	0.02	0.00	0.03	0.00	0.02	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.03	0.00	0.00	0.05	0.00	1.00