Geometric Features of Financial Markets

Anonymous Author(s)
Affiliation
Address
email

Abstract

Financial markets are complex for many reasons, one of many being the existence of an implicit structure that does not make itself immediately obvious to even the moderately-informed investor. I review classical methods in how to learn a graph structure on financial markets, as well as various elements of the design space in how to build said graphs. I then tweak the classical algorithm that utilized Minimum Spanning Trees to provide a graph that can have a variable number of edges over time, and show through Granger causality tests and a vector autoregression that geometric features provide some semblance of predictive power for the S&P500 index. I compare results to the classical MST construction and show that the variable edge algorithm, which strictly contains the MST, produces more informative graphs than the MST algorithm.

1 Introduction

Financial machine learning is a notoriously difficult field – non-stationarity, violation of IID assumptions, high probabilities of overfitting, low signal-to-noise ratios, and so many more factors make it a daunting task. Finding even a small signal-turned-feature that your competitors do not could translate into a massive advantage in live trading performance.

In this sense, analyzing one series at a time leaves much to be desired. The stock market, for example, is complex not only because each individual company engages in unpredictable activities, but because there is an implicit structure behind which everything operates. If one can understand this structure, one may be able to make better predictions and therefore better decisions. In essence, a graph structure on the market could provide additional signals to be used in predictive models (for example, using node embeddings of a market graph as a feature in predictive models to aggregate neighborhood information). I present this paper as an argument for the potential that the introduction of **geometric** (graphical) methods for financial machine learning have. The field is young and undeveloped, but shows theoretical and empirical promise.

Why might a graphical or geometric approach to market prediciton complement classical methods better than other advancements in the field? The answer lies in how graphical machine learning methods tackle the IID assumption – rather than assuming that every node has some IID probability distribution associated with it, graphical methods toss this assumption from the start! The entire point of graphical methods is to model these relationships, immediately giving them an edge over classical approaches if one takes as true the assumption that there is useful information encoded in the relationship between time series. That being said, this means that part of creating a strong graphical machine learning pipeline is creating a maximally informative graph. Determining what constitutes “maximal informativity" or what that even means is a more difficult matter that, for the most part, is beyond the scope of this paper.

Submitted to 30th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

Once a graph has been constructed, one can start applying some differential geometric techniques. A graph is, in essence, a one-dimensional approximation of a manifold. While beyond the scope of my expertise, methods from information geometry may be applicable once a graph structure has been learned [4]. The core of this project is to see if we can extract any useful geometric features from the graph itself.

Beyond the potential use cases in financial machine learning, work in this field may help shed more light on the behavior of markets directly before, during, and after periods of stress on the market such as the Great Recession of 2008. Indeed, graphs of financial markets have already been used to aid in the construction of optimal, risk-mimimizing portfolios, as there appear to be connections between the graph structure and the optimal Markowitz portfolio [5].

2 Previous Work

Much of the inspiration for this work comes from Marti et al. and their late 2020 paper summarizing the history of correlations, hierarchies, networks, and clustering in financial markets [7]. In this section, I provide a brief summary of their discussion and some of the many possible applications.

To be clear, the focus of this paper is on the stock market, though these methods could theoretically be applied to any set of time series data that follows multiple quantitative variables over time.

2.1 Background: The Standard Methodology

The most commonly adapted methodology stems from Mantenga's 1999 seminal paper [3]. For time series $1 \leq i \leq N$, let $P_i(t)$ denote the price of asset i at time t. Let $r_i(t)$ denote the difference in logged prices (henceforth referred to as *returns*) of asset i at time t, defined by

$$r_i(t) = \log \frac{P_i(t)}{P_i(t-1)} = \log(P_i(t)) - \log(P_i(t-1)). \tag{1}$$

Then we can compute the correlation ρ_{ij} between two assets i and j. The standard methodology adopts the Pearson correlation, which can be estimated as

$$\rho(x, y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}, \tag{2}$$

Where x, y represent the observations for the two series of interest, respectively, and $\bar{x}$ and $\bar{y}$ represent the average values of said time series. This measure can be negative, however, so it represents a poor choice of a distance measure between two distributions. To solve this issue, one can convert this measure into a distance d_{ij} via the following formula:

$$d_{ij} = \sqrt{2(1 - \rho_{ij})}, \tag{3}$$

which satisfies all necessary properties of a metric.

With these distances in hand, one can build a graph with nodes representing the individual assets, and edges between them representing distances between their corresponding time series. We do so via a **Minimum Spanning Tree (MST)**. One of many algorithms for doing this is listed below:

This algorithm constructs the graph with minimum possible total edge weights such the graph is still fully connected. Each node is connected to at most two other nodes – the number of edges is fixed. Note the implicit introduction of some exogenous parameters here: N, the number of nodes (series) we are considering, and T – the length of the sliding window used to generate the graph. A sliding window is important to capture how the graph changes over time. For sliding windows of length T, we generate $N - T$ graphs, one for each possible sliding window over the graph. Note that this

Algorithm 1 Kruskal's Algorithm

```
function MST({d_ij}_{i,j=1}^N)                    ▷ Initialize a fully disconnected graph G = (V, E)
    E ← ∅
    V ← (i, j)_{i,j=1}^N                          ▷ Try to add edges by increasing distance
    D ← {(d_ij), r}          ▷ D is the ordered list of distances, with r corresponding to rank
    while G not connected do                      ▷ Check that (i, j) are not connected
        if not connected(i, j) then
E ← E ∪ {(i, j, weight = d_ij)}                   ▷ Add the edge connecting nodes i, j
return G = (V, E)
```

can be incredibly computationally expensive, as the graph generation algorithm is at least $O(N^2)$ in computing the distances (assuming the distance calculation has constant time complexity in N, which it usually is not).

Furthermore, it should be noted that there is another implicit dependency in the **sampling frequency**. This will be addressed in a future section, but the researcher must make a choice between operating on an intraday, daily, weekly, monthly, etc. time scale. It is likely that the evolution of the graph in time will change based on the time scale, but exactly how remains an open question.

2.2 Using a Graph

Now, once one has a a graph, one can study various things, including but not limited to

- Studying regime shifts using graphical features
- The dynamics of the graph using summary statistics such as the survival ratio of the edges, node degree, strength, centrality measures (eigenvector, betweenness, and closeness), and the agglomerative coefficient.
- Where the optimal portfolio lies in the graph
- How the size of the graph changes over time.

2.3 Ollivier-Ricci Curvature and Market Fragility

Previous work by Sandhu et al. [8] generalizes the idea of Ricci curvature to a discrete space (graph), where it is then called the **Ollivier-Ricci Curvature**. The Ollivier-Ricci curvature is calculated as

$$\kappa(x, y) = 1 - \frac{W_1(\mu_x, \mu_y)}{d(x, y)}, \tag{4}$$

where W_1 represents the Wasserstein-1 metric, also known as the Earth Mover's Distance, between probability distributions μ_x and μ_y. These distributions are defined as vectors $\mu \in \mathbb{R}^N$ such that $\mu_x(y)$, the y-th element of μ_x, is the probability that a random walker starting at node x will end up at node y. This can be calculated as

$$\mu_x(y) = \frac{w_{xy}}{d_x}, \tag{5}$$

where

$$d_x = \sum_{y \in \mathcal{N}(x)} w_{xy} \tag{6}$$

is the sum of all weights among its neighbors. The astute reader can gather that these are simply columns of a transition matrix, where the matrix is calculated from the graph's adjacency matrix and columns are normalized to sum to 1. That is, if T is the transition matrix obtained via this method, then μ_x is simply the column of T corresponding to node x.

The importance of the Ollivier-Ricci curvature (henceforth referred to as curvature), as derived in the paper, is that it is negatively correlated with market fragility. As unintuitive as it seems, this translates to an increase in curvature correlating with a large deviation from a stationary distribution. That is, a market crash or boom showing large changes will be associated with an increase in average curvature of the graph.

Sandhu et al. find that their experimental data (mentioned in some more detail later) show that Ollivier-Ricci curvature tends to increase during times of crisis. Additonally, they find that global network entropy, a nodal feature that is correlated with fragility, is positively correlated with curvature (an edge feature).

3 The Design Space

With the previous allusion to three implicit choice parameters (the number of assets N, the sliding window length T, and the sampling frequency of the time series), we explore some more of the options for tweaking how the graph is generated. Unless otherwise stated, most of the information from this section was extracted from Marti et al. [7].

3.1 Motivation for Exploring the Design Space

While the method above is the standard, widely accepted method, Marti et al. point out some reasons why it may be less than optimal:

- Clusters obtained from the MST are notoriously unstable and are heavily subject to perturbations in the onput data
- The Pearson linear correlation is notoriously brittle to outliers and do not represent the relationship between non-Gaussian distributions well
- There are no theoretical results showing the statistical reliability of correlation-based networks.
- A change in the (admittedly arbitrary) method of constructing the graph can have huge implications for the outcomes. As such, any implications of optimal portfolio construction or other graph-centric results in finance

All of these issues warrant an exploration into alternative design options that may not run into problems of this nature.

3.2 Clustering Algorithms

The following are some of the different variants of algorithms that can be used to construct the graph:

- Average Linkage Minimum Spanning Tree (ALMST): Helps remedy the unwanted chaining phenomenon of the MST.
- Planar Maximally Filtered Graph (PMFG): Strictly contains the MST, but encodes a larger amount of information in its internal structure
- Maximum likelihood: define a likelihood function and find a clustering with high likelihood

3.3 Choice of Metric

The Pearson correlation, which leaves the option to define the metric in alternative ways:

- Use a more robust-to-outliers correlation measure such as the Spearman correlation

- Quantify the amount of information one series provides about another with measures such as Granger causality and partial correlation
- Capture non-linear relationships via information-theoretic, copula-based, and tail dependence distances

3.4 Preprocessing of Time Series

Oftentimes, when considering data for tickers, one needs to find ways to control for the effect of the entire market. For example, if the stock market itself is performing well, then every stock will perform well. Thus, if we want to understand how well stocks are doing relative to each other, we may want to remove the market effect. This can be done in many ways:

- Include the market index as a node in the graph
- Using dynamic correlations using a Dynamic Spanning Tree (DST) [9]
- Regressing each ticker's returns on the market index and subtracting out the market index's effect on the ticker returns.

3.5 Sampling Frequency

Any time one works with time series data, sampling frequency matters. As mentioned earlier, the behavior of a time series is largely dependent on the sampling frequency. We are likely to see different behaviors on 1-minute time scales, 1-hour time scales, and 1-day time scales.

However, questions have been raised as to whether or not storing data by time is actually an effective way to do financial data analysis. Authors such as Marcos Lopez de Prado [6, Chapter 2] argue for the usage of alternative "bar types" – that is, alternative ways of sampling prices. In the interest of brevity, I invite the reader to check out the reference for a detailed description of alternative bar types.

The basic gist of it, however, is that time-sampled series for stocks are not the most effective way to represent data. Stocks are typically traded more in the beginning and end of a day, and tend to enter a lull period in the middle. Thus, by sampling at constant time intervals, one is effectively under-sampling during high trading periods and oversampling during low trading periods. Ideally, we would like to sample as information arrives to the market. Lopez de Prado writes about some ways to do this for single tickers, but this can be generalized by summing statistics and sampling all tickers at once. Some possibilities are:

- Volume Bars: Sampling after a certain volume is traded
- Money Bars: Sampling after a certain dollar value is traded (to account for varying prices over time)
- Information Imbalance Bars: Sampling as informed traders arrive to the market (there is an imbalance of sell vs buy orders placed on the order book)

Unfortunately, computing these bars requires access to tick data (individual trades) and the order book for each company, which is unavailable for free.

4 Approach of this paper

Given the size of the design space (of which I have only presented a small portion of), I have to choose carefully which elements I am able to explore.

Since the method of constructing the tree is somewhat arbitrary, I choose to compute both the MST and a graph larger than the MST (which strictly contains the MST). The reason for this differentiation is that I find it unrealistic that each node should be connected to only two other nodes, and imposing that restriction without any a priori knowledge of what a financial graph *should* look like seems ineffective. However, making a claim either way without any evidence is not good science – I choose to compute both and compare the results empirically.

To generate the larger graph, I have to make a simple modification to Kruskal's algorithm listed above: rather than connecting nodes i and j if they are not *connected* via some path in the graph, I connect them if they are not explicitly connected by an edge (within a 1-hop neighborhood). Thus, to construct the graph, I continuously add edges in order of increasing weight (ignoring 0) until every node is a part of the graph (i.e., the graph is connected). Rather than limiting the graph to $N-1$ edges, the number of edges in this formulation can change over time (with a minimum of $N-1$ possible edges, representing the MST). Thus, based on current market conditions, a pair of stocks may have their connection adjust in weight or even break their link according to the data. It is worth mentioning that I assign edge weights via these distances, and as such I use a weighted graph in my analysis. I refer to graphs generated by this algorithm as Dynamic Edge Count graphs, henceforth denoted as DEC graphs.

As a distance metric, I use the Spearman correlation due to aforementioned issues with the Pearson correlation being brittle due to outliers. Once I obtain the Spearman correlation matrix, I still use the same conversion to distance listed in Equation 3. I considered using the Brownian Distance Correlation [10], but had to decide against it due to its large computing runtime making it infeasible to compute on my available hardware in the given timeframe.

The data I use are daily data for stocks listed on the S&P500 that have existed since January 1, 2005 up until October 20th, 2021. The reason for this is I was interested in the behavior of the graph before, during, and after the financial crises of both the Great Recession and the COVID-19 pandemic (as well as other crises, such as the European Debt Crisis). This totals 198 stocks, leaving a total of about $19,000$ possible pairs. I pulled this data from Yahoo Finance using the yfinance Python wrapper.

I started with a rolling window of $T = 25$, since I wanted my window to be slightly larger than the average number of trading days in a month (21, with a max of 23 in March). This leaves a total of about $4,400$ graphs. To check the robustness of the results at larger time periods, I also compute graphs on a rolling window of $T = 125$ days.

While playing with alternative bar types seemed enticing to me, I was unable to source the data without paying large amounts for a stock data API. Additionally, free intraday data only go out about 60 days, and so analyzing a dataset constructed of intraday data would not allow me to analyze any behavior during financial crises (once again, this could be remedied with paid data).

With this graph in mind, I attempt to reproduce the work of Sandhu et al. on the Ollivier-Ricci curvature of a market graph. They used an MST on rolling windows of size $T = 22$ and $T = 132$ days using the Pearson correlation-turned-metric transformation detailed in equation 3. Additionally, they use price data rather than log returns as detailed by Equation 1. The interesting question here, then, is how exactly the results of Sandhu et al.'s paper change when we change position in the design space?

The main question of this paper, however, is: are geometric features of financial markets helpful in providing any forecasting information? We investigate this question through two methods: Granger causality tests [1] as well as a vector autoregression (VAR) analysis.

The Granger causality test is a statistical test between two time series that helps in determining if one time series is useful in forecasting another. It is worth mentioning that it is not a symmetric test – the p-value for $G(X, Y)$ may not (and usually will not) be the same as the p-value for $G(Y, X)$. For parameters (X, Y), we denote $G(X, Y)$ to be the Granger causality test to see if Y "G-causes" X. If $p < 0.05$, we say that Y G-causes X and is therefore useful in forecasting X.

In terms of features used, we examine a (very small) subset of possible graph features (averages taken over all nodes):

- Average edge weight
- Average clustering coefficient (DEC graphs only)

- Average Ollivier-Ricci curvature
- Graph density (DEC graphs only)
- Number of edges over time (DEC graphs only)

So, in a similar fashion to how I transformed stock price data, I compute log-returns for the S&P500 market index and determine which of these features provides statistical significance in a Granger causality test up to 5 lags (5 is an arbitrary choice). Once the Granger causality tests are complete, I fit a VAR model of order 5 regressing the S&P500 index on average edge weight, average clustering coefficient, average curvature, and graph density (as well as lags of the S&P500 index itself), and examine the statistical significance of the features to see if we can observe clear signs of geometric features being useful in forecasting efforts.

5 Results

The results of my experiment are shown in Figures 1 to 9. The shaded areas, from left to right, represent the Great Recession (with the market crash happening on March 6, 2009), the European Debt Crisis (starting in July 2010, with a particularly bad spike in summer of 2011), and the COVID-19 drop, which started in March 2020. There may be other financial crises that are not shaded. These crises simply serve as benchmarks. Every point in the upcoming figures represents a graph trained on a sliding window of length T, with the point corresponding to the end date of the sliding window (what a researcher would have as a constructed graph at the end of a trading day).

Examining figure 1, we see that the curvature of the DEC graphs tend to spike near times of financial collapse. Interestingly enough, there is some evidence that curvature may be a leading indicator for a crash, as it tends to spike days before a large crash (look at the 2009 drop, for example – curvature had already spiked in February, while the crash did not occur until March 6th). I examine this later with rigorous statistical testing later in the paper. Note that the MSTs exhibit very little variation over time with respect to curvature. This is also true of their edge weights. Since they are constructed with the minimal possible distances, it tends to be the case that their average edge weight does not change much. Immediately we see the difference between my results and Sandhu et al. – while the DEC graphs have comparable behavior in that curvature tends to spike in times of crisis, the MSTs show no considerable variation in average curvature during crisis times. This may be due to several factors:

- I used the Spearman correlation rather than the Pearson correlation
- Sandhu et al. considered only curvatures for which $d(x, y) = 1$ for two nodes x and y. Additionally, they used an unweighted graph, whereas I used a weighted graph.

While the results of this paper do not exactly match those of Sandhu et al.'s work with respect to the behavior of curvature, one can at least notice the similarities for the DEC graphs: during times of financial crisis, the curvature tends to increase drastically. This confirms the derivation that curvature is negatively correlated with fragility (and therefore positively correlated with large changes in prices).

Moving on to average clustering coefficients (Figures 2 and 3), visual inspection does not necessarily show any obvious significant variation for $T = 25$, but one could argue there is some relationship between market crashes and clustering for the $T = 125$ graphs.

Examining graph density, we see that the graph tends to be more dense in times of crisis (Figures 4 and 5). This corresponds with previous theory that stocks tend to move together in times of crisis, so the number of edges should increase accordingly. We, of course, see the exact same behavior in Figures 6 and 7, since density and increasing function in the number of edges, and the number of nodes stays constant. These plots all illustrate an expected point, but a point nonetheless – graphs learned on longer time frames tend to have much smoother feature changes over time than graphs learned on shorter time frames. However, while larger values of T tend to smooth out features and make them more resistant to noise, they also delay the response of a change in these variables to market conditions, We particularly see this in Figures 4 and 5. In Figure 4, the peak density during the 2009 crash occurs during the crisis. However, in Figure 5, the $T = 125$ graphs do not see peak density until a few periods *after* the crisis is over.

From these graphs, one may be able to deduce the existence of another crash in the time period 2015-2016 due to large spikes in curvature, density, and the number of edges. Indeed, there was another crash in this time period known as tbe 2015-2016 stock market selloff, during which over $10 trillion were wiped off of global markets.

This exposition into graph features evolving over time provide a clear indication that geometric features of the market encode *some* information. Whether or not it is useful is another question – and the subject of the Granger Causality tests. The results of this test are shown in Table 1. Recall that a statistically significant result (denoted $**$ for significance at the 0.05 level and $***$ for significance at the 0.01 level) means that the variable is useful in forecasting the value of the target series (S&P500 difference in log prices) up to some number of lags. Key takeaways of the results are as follows:

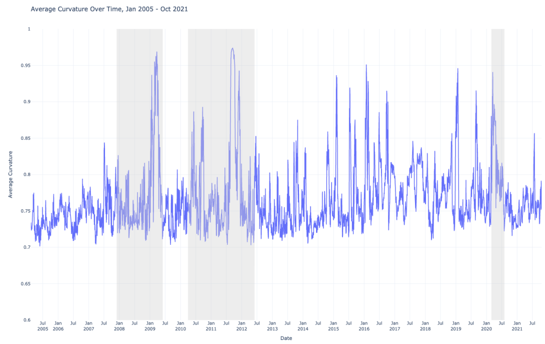
- Average curvature tends to be somewhat useful in forecasting for the $T = 25$ graphs.
- The $T = 125$ graphs see the most amount of useful features at the highest levels of significance: graph density, average edge weight, and average curvature. The clustering coefficient does not prove to be useful.
- None of the available features (average edge weight and average curvature) for MSTs are useful in forecasting
- DEC graphs tend to encode more useful information for forecasting than MSTs

The results of the Granger causality tests are indicative of an interesting relationship – graphs with smaller values of T tend to be more sensitive to noise, and larger values of T tend to make graph features respond late to market events – but their features are more useful in prediction. This prompts the question of if there is an optimal value of T such that forecasting information and sensitivity to noise are optimally balanced. Since T is a hyperparameter, one would have to turn to hyperparameter optimization methods in order to investigate this question.

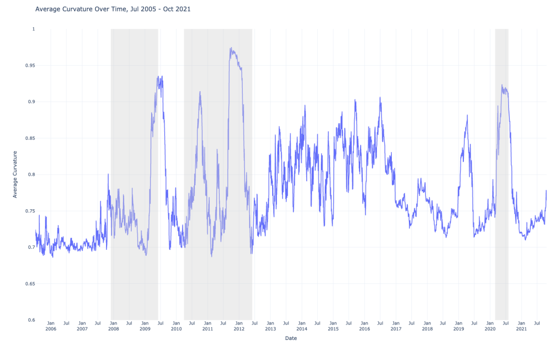
With the results of the Granger causality tests in mind, I decided to give a first attempt at a VAR to see what variables I could find statistical significance for. I was less-so concerned with the R^2 value (hovering around $3 - 3.5\%$ for each model) that I was with seeing which coefficients we could find to be statistically different from 0. The full regression results are shown in Table 2, but a quick synopsis of the significant variables at the 0.05 level or lower are listed here:

- $T = 25$ DEC: Average edge weight (lagged 4 periods), average clustering coefficient (lagged 1 period), graph density (lagged 1 and 4 periods), and the S&P500 difference in log prices (lagged 1, 4, and 5 periods).
- $T = 125$ DEC: Average edge weight (lagged 1 and 5 periods), graph density (lagged 1 period), and the S&P500 difference in log prices (lagged 1, 4, and 5 periods).
- $T = 25$ MST: Average curvature (lagged 3 periods) and the S&P500 difference in log prices (lagged 1, 4, and 5 periods).
- $T = 125$ MST: Average edge weight (lagged 5 periods) and the S&P500 difference in log prices (lagged 1, 4, and 5 periods).

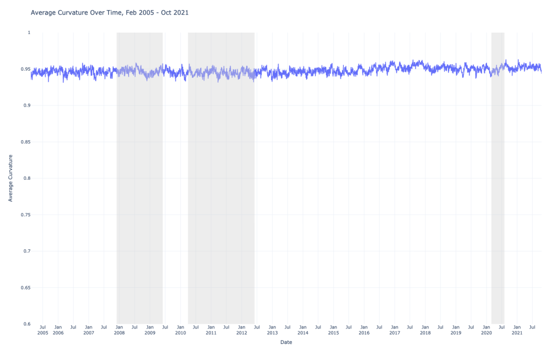
One might notice that these results suggest something different than what the Granger causality tests might. Very likely, the issue is that a Granger causality test is computed by considering only two series at a time – the target, and the variable we want to examine the explanatory power of. When including all features in a regression, we introduce multicollinearity effects in the regression itself, likely taking away from the significance of some factors that were expected to be significant, and providing some spurious results (e.g., clustering's significance for the $T = 25$ DEC graph).



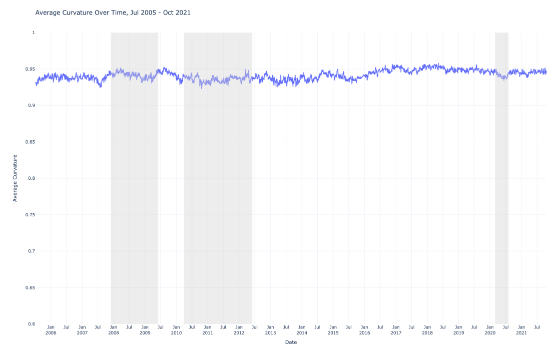
(a) $T = 25, N = 198$, DEC Graphs



(b) $T = 125, N = 198$, DEC Graphs



(c) $T = 25, N = 198$, MST



(d) $T = 125, N = 198$, MST

Figure 1: Average Ollivier-Ricci curvature for all 4 generated graphs over time. Note that the Dynamic Edge Count graphs exhibit much more variation in their curvature over time than the MSTs.

Figure 2: Average Clustering Coefficient Over Time, $T = 25, N = 198$

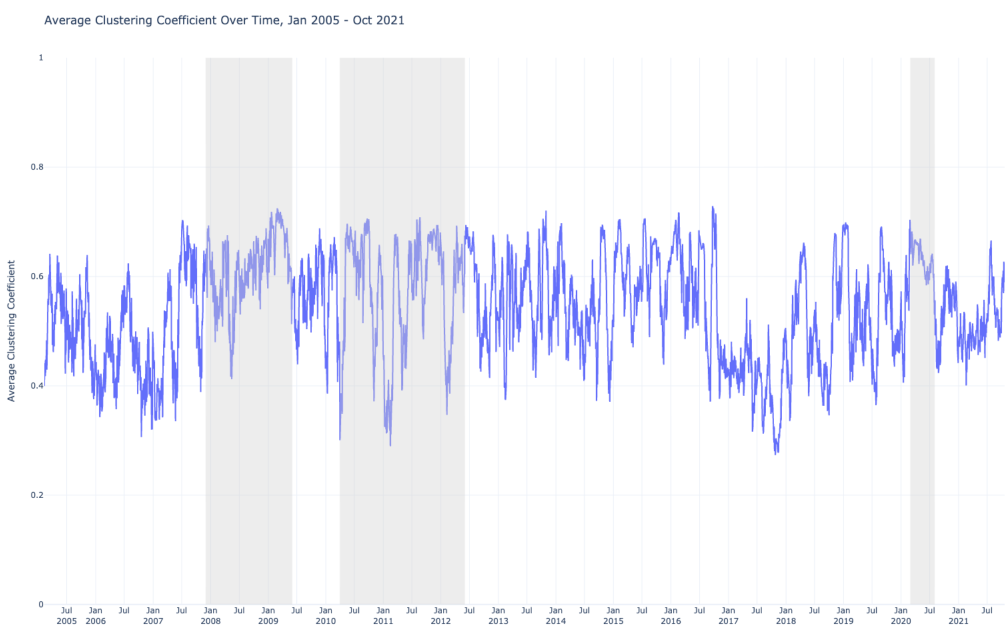


Figure 3: Average Clustering Coefficient Over Time, $T = 125, N = 198$

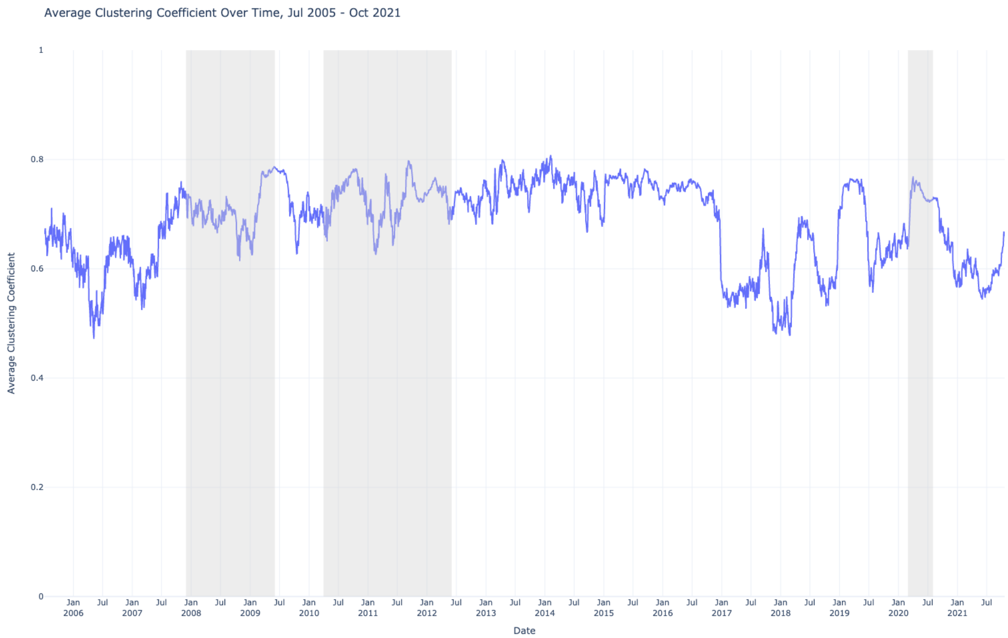


Figure 4: Graph Density Over Time, $T = 25, N = 198$

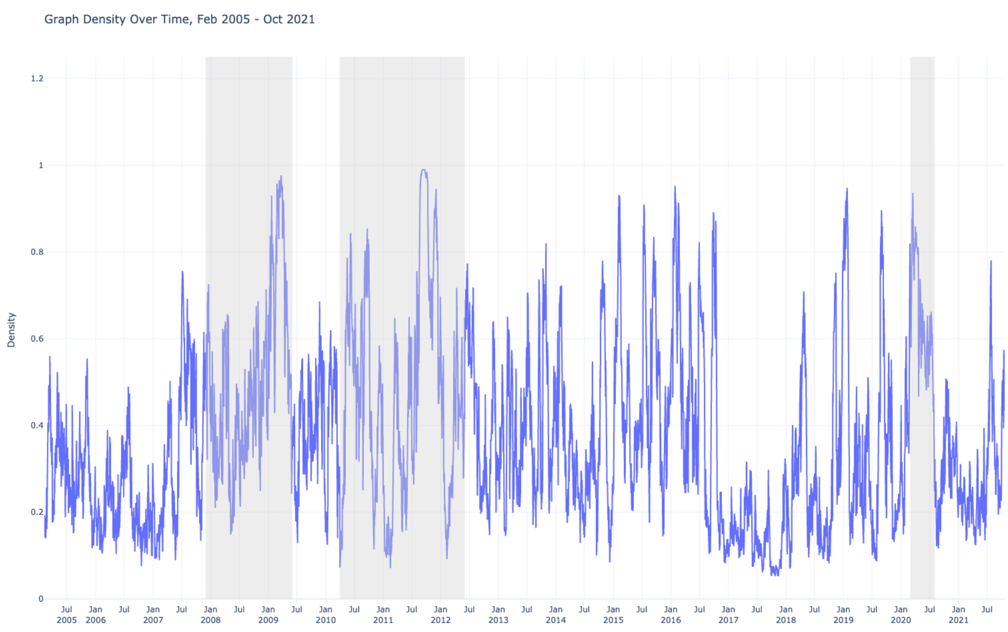


Figure 5: Graph Density Over Time, $T = 125, N = 198$

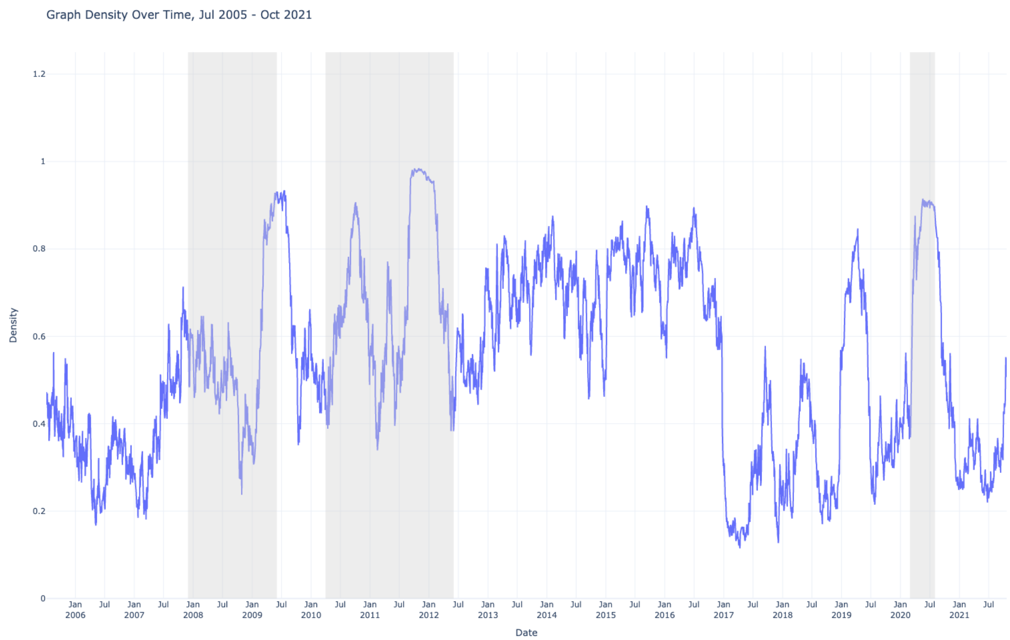


Figure 6: Number of Edges Over Time, $T = 25, N = 198$

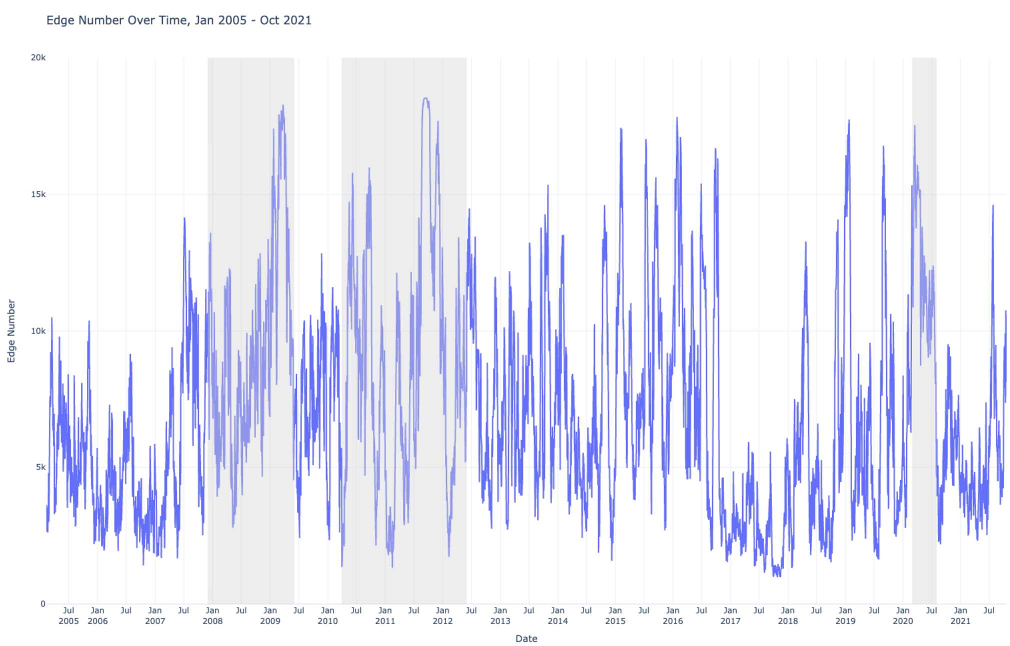


Figure 7: Number of Edges Over Time, $T = 125, N = 198$

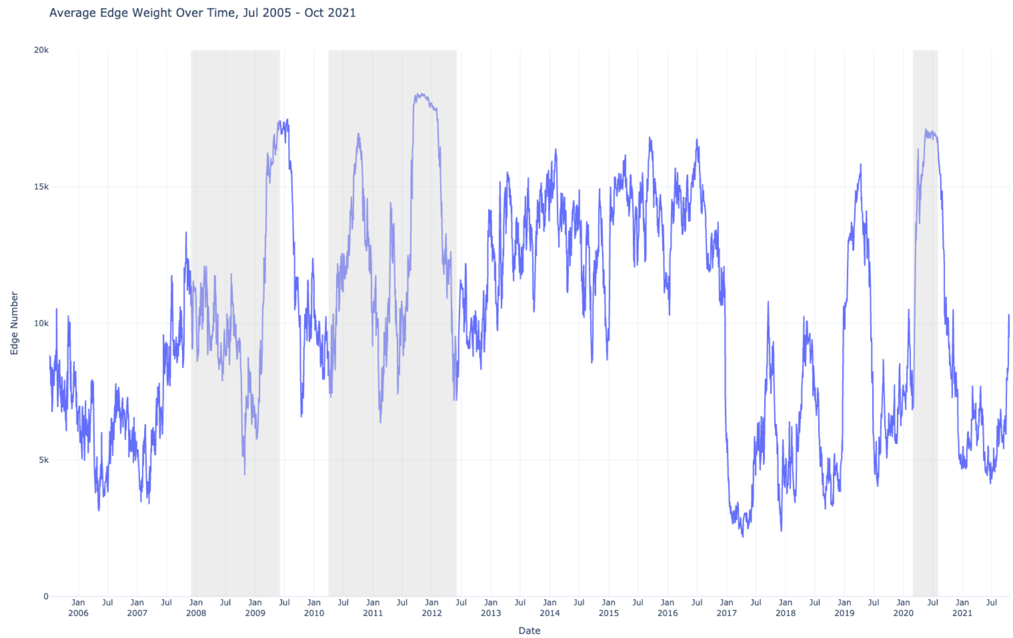


Figure 8: Average Edge Weight Over Time, $T = 25, N = 198$

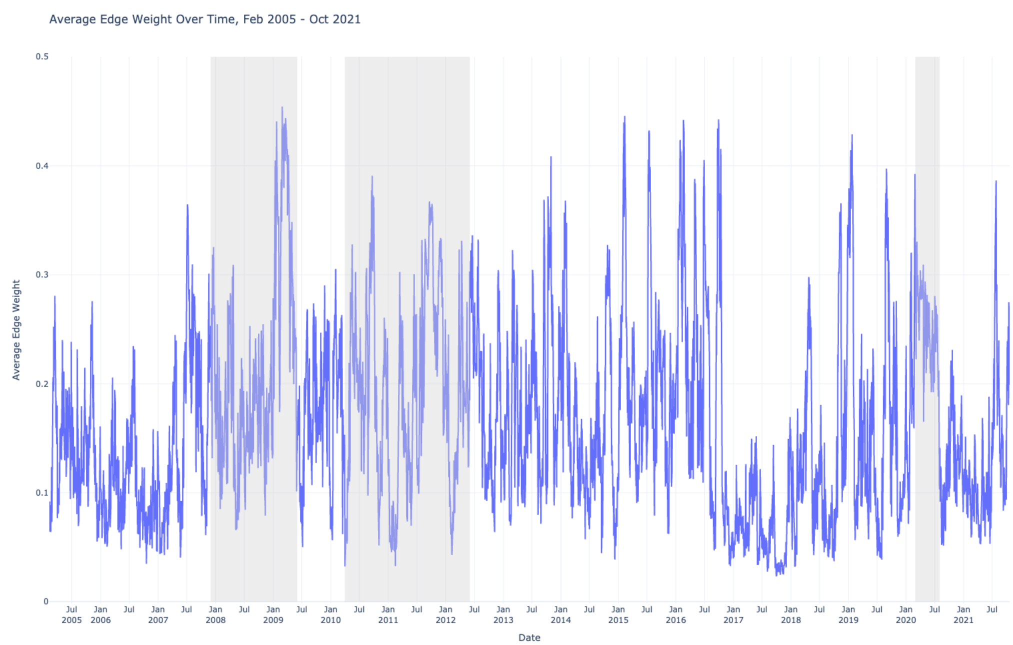


Figure 9: Average Edge Weight Over Time, $T = 125, N = 198$

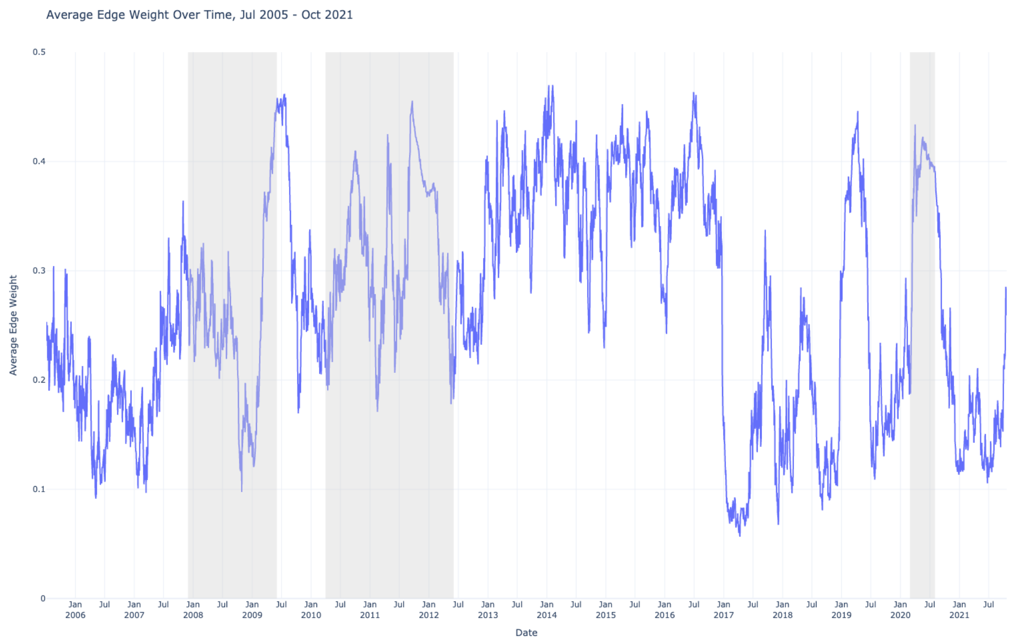


Table 1: Granger Causality Test Results

Graph Type	Variable Name	lag_1	lag_2	lag_3	lag_4	lag_5
spearman_25	density	0.070	0.072	0.136	0.228	0.217
	avg_clustering	0.267	0.458	0.640	0.799	0.850
	average_weight	0.051	0.083	0.160	0.269	0.249
	avg_curvature	0.028**	0.045**	0.085	0.088	0.062
spearman_125	density	0.003***	0.008***	0.025**	0.023**	0.037**
	avg_clustering	0.063	0.167	0.274	0.398	0.518
	average_weight	0.003***	0.011**	0.032**	0.035**	0.053
	avg_curvature	0.000***	0.001***	0.002***	0.002***	0.003***
spearman_25_mst	average_weight	0.437	0.655	0.807	0.885	0.921
	avg_curvature	0.377	0.611	0.772	0.529	0.640
spearman_125_mst	average_weight	0.139	0.175	0.256	0.253	0.334
	avg_curvature	0.961	0.561	0.501	0.641	0.630

6 Conclusion and Future Work

The results of this experiment are promising. From our visual inspections, we have seen that we can construct meaningful geometric representations of financial markets. From the Granger causality tests, we have seen that geometric features are useful in providing forecasting features, and these features can be woven into machine learning pipelines. From the VAR analysis, we have seen that some of these geometric features can provide statistical significance – meaning they provide some sort of explanatory power in describing why the market index changes in the way it does.

While the results of Sandhu et al. were not replicable with respect to the MSTs (possibly due to some of the aforementioned reasons), I was able to replicate the results using a completely different graph generation algorithm and different metric: some geometric features do tend to vary significantly from the average during periods of financial crashes.

While the exposition may not be completely polished, this paper provides promising results for more complex methods to be tested. The current methodology considers only graph-level features, some of which are obtained by averaging node-level and edge-level features. A better pipeline may want to retain these features and apply a more advanced algorithm that allows them to be taken into account, such as GraphSAGE [2]. Certainly, this type of research is interesting and could bear many fruits for a financial machine learning practitioner and/or a quantitative trader. It has been noted that Ricci curvature can be a useful indicator, for example, in mean reversion strategies [5].

However, another avenue exists. These results exist for one point in the design space of a graph. If we are to make any claims of robustness and worth of geometric features of markets, it is important to check different parts of the design space. One could endeavor to test these results for different metrics and different graph generation algorithms, as well as adjusting the rolling window length. Metrics that can capture nonlinear relationships, while computationally expensive, may be the key to seeking out truly informative features. Alternatively, with the graphs that have been constructed, one may attempt to perform even more feature engineering, as well as playing around with different transformations of the response variable. A particularly interesting topic in terms of differencing is the discussion on fractional differentiation in [6].

Furthermore, one could consider yet another path. The current methodology involves computing statistics for an entire sliding window. However, within this window, the dynamics of the graph are largely ignored. That is, we don't tend to construct a graph using the changes in value for each of the nodes between each time step – rather, we just compute a metric over the whole period. This may not be optimal. In my senior thesis, I am currently investigating if we can construct a dynamical model that learns an adjacency matrix based on a nonlinear Gaussian noise model that takes into account the evolution of the graph at each time step of the window. Specifically, the learned adjacency matrix is one that best explains the time evolution of the system over the duration of the entire rolling window.

Regardless of the path to move forward, it is promising that the next large breakthroughs in financial machine learning will heavily rely on the usage of graph theory and graph machine learning methods. For a complex structure such as a stock market, it is only natural that we will need to model relationships – and work to this point shows that this may be the correct language for doing so.

Table 2: SP500 VAR Results

	Graph Type Used to Generate Features			
	Window 25	Window 125	Window 25 (MST)	Window 125 (MST)
average_weight_lag1	-0.018	0.079**	-0.088	3.447
	(0.012)	(0.040)	(0.892)	(3.628)
average_weight_lag2	0.012	-0.082	-0.641	-6.359
	(0.017)	(0.056)	(1.270)	(5.157)
average_weight_lag3	0.030*	-0.041	0.426	-1.564
	(0.017)	(0.056)	(1.272)	(5.171)
average_weight_lag4	-0.046***	-0.043	-1.199	-3.222
	(0.017)	(0.056)	(1.270)	(5.156)
average_weight_lag5	0.021*	0.086**	1.147	7.369**
	(0.012)	(0.040)	(0.885)	(3.605)
avg_clustering_lag1	-0.013**	0.008		
	(0.006)	(0.014)		
avg_clustering_lag2	0.012	-0.010		
	(0.008)	(0.018)		
avg_clustering_lag3	0.000	-0.004		
	(0.008)	(0.018)		
avg_clustering_lag4	0.001	0.023		
	(0.008)	(0.018)		
avg_clustering_lag5	-0.001	-0.021		
	(0.006)	(0.014)		
avg_curvature_lag1	-0.011	0.013	0.018	0.040
	(0.008)	(0.015)	(0.017)	(0.027)
avg_curvature_lag2	0.012	-0.030	0.009	-0.022
	(0.010)	(0.020)	(0.020)	(0.031)
avg_curvature_lag3	-0.003	0.015	-0.041**	-0.018
	(0.010)	(0.020)	(0.020)	(0.031)
avg_curvature_lag4	0.004	0.007	0.014	-0.030
	(0.010)	(0.020)	(0.020)	(0.031)
avg_curvature_lag5	-0.000	-0.002	0.001	0.031
	(0.008)	(0.015)	(0.017)	(0.027)
density_lag1	0.018***	-0.050**		
	(0.007)	(0.023)		
density_lag2	-0.014	0.060*		
	(0.009)	(0.032)		
density_lag3	-0.015	0.020		
	(0.009)	(0.032)		
density_lag4	0.023**	0.013		
	(0.009)	(0.032)		
density_lag5	-0.010	-0.042*		
	(0.007)	(0.023)		
sp500_lag1	-0.086***	-0.090***	-0.087***	-0.087***
	(0.011)	(0.011)	(0.011)	(0.011)
sp500_lag2	-0.011	-0.018*	-0.012	-0.014
	(0.011)	(0.011)	(0.011)	(0.011)
sp500_lag3	0.009	0.008	0.009	0.009
	(0.011)	(0.011)	(0.011)	(0.011)
sp500_lag4	-0.027**	-0.030***	-0.027**	-0.027**
	(0.011)	(0.011)	(0.011)	(0.011)
sp500_lag5	-0.039***	-0.041***	-0.038***	-0.038***
	(0.011)	(0.011)	(0.011)	(0.011)
Observations	4,199	4,099	4,199	4,099
R^2	0.033	0.035	0.031	0.031
Residual Std. Error	0.055(df = 4174)	0.055(df = 4074)	0.055(df = 4184)	0.055(df = 4084)
F Statistic (df)	7.07*** (14; 4174)	7.06*** (14; 4074)	10.8*** (15; 4184)	10.7*** (15; 4084)

Note: *p<0.1; **p<0.05; ***p<0.01

References

[1] C. W. J. Granger. "Investigating Causal Relations by Econometric Models and Cross-spectral Methods". In: *Econometrica* 37.3 (1969), pp. 424–438. ISSN: 00129682, 14680262. URL: http://www.jstor.org/stable/1912791.

[2] William L. Hamilton, Rex Ying, and Jure Leskovec. *Inductive Representation Learning on Large Graphs*. 2018. arXiv: 1706.02216 [cs.SI].

[3] R.N. Mantegna. "Hierarchical structure in financial markets". In: *The European Physical Journal B* 11.1 (Sept. 1999), pp. 193–197. ISSN: 1434-6028. DOI: 10.1007/s100510050929. URL: http://dx.doi.org/10.1007/s100510050929.

[4] Frank Nielsen. "An Elementary Introduction to Information Geometry". In: *Entropy* 22.10 (2020). ISSN: 1099-4300. DOI: 10.3390/e22101100. URL: https://www.mdpi.com/1099-4300/22/10/1100.

[5] J.-P. Onnela et al. "Dynamics of market correlations: Taxonomy and portfolio analysis". In: *Physical Review E* 68.5 (Nov. 2003). ISSN: 1095-3787. DOI: 10.1103/physreve.68.056110. URL: http://dx.doi.org/10.1103/PhysRevE.68.056110.

[6] Marcos Lopez de Prado. *Advances in Financial Machine Learning*. 1st. Wiley Publishing, 2018. ISBN: 1119482089.

[7] "Progress in Information Geometry". In: *Signals and Communication Technology* (2021). ISSN: 1860-4870. DOI: 10.1007/978-3-030-65459-7. URL: http://dx.doi.org/10.1007/978-3-030-65459-7.

[8] Romeil Sandhu, Tryphon Georgiou, and Allen Tannenbaum. *Market Fragility, Systemic Risk, and Ricci Curvature*. 2015. arXiv: 1505.05182 [q-fin.RM].

[9] Ahmet Sensoy and Benjamin Tabak. "Dynamic spanning trees in stock market networks: The case of Asia-Pacific". In: *Physica A: Statistical Mechanics and its Applications* 414 (Nov. 2014), pp. 387–402. DOI: 10.1016/j.physa.2014.07.067.

[10] Gábor J. Székely and Maria L. Rizzo. "Brownian distance covariance". In: *The Annals of Applied Statistics* 3.4 (2009), pp. 1236–1265. DOI: 10.1214/09-AOAS312. URL: https://doi.org/10.1214/09-AOAS312.