Topological Data Analysis in Quantitative Finance: An Adaptative Model

Patrick F. Norton, Jack. B Jedlicki

ABSTRACT

In this study, we build upon previous research conducted on the use of topological data analysis of time series data, using an approach based on persistence homology. We develop a first model using persistence homology as reported in the literature, e.g. by Gidea et al., and extend the analysis to include daily returns of Hilton Hotels Corporation Common Stock during the COVID-19 pandemic crash of 2020. Our findings indicate that Wasserstein distances exhibit significant growth before the primary peak of a crash in the vicinity of a financial meltdown. However, a deeper look into the method reveals that the approach is indeed flawed since it relies on using data of the days involved in the crash. Hence, we propose a second persistence homology model, representing a novel approach that aims to predict crashes of stock market value, with daily prices known only up to the day before the crash. Our model, so-called adaptative persistence homology, improves the persistence homology-based analysis by providing the optimal parameters of the persistence homology model for predicting the variations of the stock with maximum accuracy. Hence, we provide for the first time a persistence homology approach that adapts itself to each different asset in order to improve the prediction of crashes of the stock market.

1. Introduction

Topological Data Analysis (TDA) is a realm within data analytics that deals with a combination of computational, statistical and topological methods used to explore the topological features of complex and noisy multi-dimensional data-sets. TDA has proven to be a powerful tool in machine learning, neuroscience and image processing as it gives a strong analytical insight into multi-dimensional behaviours, which are also typically seen within quantitative finance.

Within finance, the general principle underlying TDA is persistence homology (5). Detailed and rigorous descriptions of the principles underlying persistence homology can be found elsewhere (4; 2; 3). The method is especially robust under noisy data, which is indeed a property given by the central theorem of persistence homology (1), stating that persistence diagrams are stable under small deformations of input data. Thus, it is interesting to test how this data analysis technique performs in noisy financial time series. In fact, the effectiveness of the topological method has been proven recently in (5), where the authors claimed a large increase in the Wassertian distances before a market crash when looking at the W-distances of a 4-dimensional space composed of the S&P 500, DJIA, NASDAQ, and Russel 2000.

The main idea of the implementation of persistence homology in stock market data is as follows. A sliding window of a fixed size is used (e.g. 100 timepoints), and for each window, a persistence diagram is computed. A persistence diagram is a two-dimensional representation of the topological features of a point data cloud. In order to construct a persistence diagram, a large number of simplicial complexes is computed from the original points by varying a scaling parameter over a certain range. Each simplicial complex contains k-dimensional holes, which can appear and disappear when varying the sliding parameter. Holes that appear and disappear relatively fast can be considered as noisy information, while holes that persist relatively long when varying the parameter represent an insightful feature of the data cloud —and hence the name *persistent* homology. In order to summarize all the information, a two-dimensional diagram representing all the holes having appeared in the set of simplicial complexes is built. The x-axis represents the birth value of the sliding parameter for each hole, while the y-axis represents the death value of the parameter. Hence, the persistent and meaningful points —representing holes that persisted for a large interval of the scaling parameter— lie far away from the $x = y$ line. The described diagram is the *persistence diagram.*

Going back to our initial time-series stock market data, recall that for each time window (containing n data points), a persistence diagram is computed, containing information about the topological features (e.g. connected components, loops, holes) of the selected point cloud. Now, the key point of persistence homology is that it has been proven in the vicinity stock market crashes, the variations of persistence diagrams over time strongly increase (5) — compared to non-crash times. One can quantify this variation of persistent diagrams over different data clouds with distances such as the L^1 or L^2 norm (5), or with the Wasserstein distance (3). In summary, in the persistence homology-based analysis of stock market data, a sliding window is generated over the time-series data, and for each time window a persistence diagram of the data cloud is computed, containing meaningful topological information about the data points. Then, the distance between consecutive persistence diagrams is computed, and large increases in the distance are used as indicators of an imminent stock value crash.

In the following, we test the method with our own implementation of the method, setting different time window sizes and different overlaps between windows, and using the Wasserstein distance. We then encounter a problem in Fig.2 with the cutoff of time series, which leads us to improve the method with the use of an adaptative persistence homology technique, called Model 2, representing an unprecedented and novel approach. We finally test the new model in data from Hilton Hotels Stock, ˆFVX and Amazon.

2. Model 1: Persistence homology

In this first step we use a method similar to the one performed in (5). The algorithm is written in Python, using functions from ripser and persim. Briefly speaking, the method we employ is the following.

1. Data extraction: We use time series data from three assets; Hilton Hotels Corporation Common Stock (HLT), Treasury Yield 5 Years (ˆFVX), and Amazon (AMZN). We have chosen these three assets for having:

    (a) Three assets in different economic contexts with different dependence over time of the topological features.

    (b) Three assets presenting remarkable crashes in the last five years (in order to try to predict these crashes with TDA).

    We take the data from Yahoo Finance, and for each asset we take their closing market prices from 2018-01-01 to 2022-04-21, with one data point per day. Next, we convert the data arrays to arrays with the natural logarithm of the ratio of the price over consecutive days (stored as r in the code). More exactly, we convert each stock price P_i on day i to the value r_i:

    $$r_i = \ln\left(\frac{P_{i+1}}{P_i}\right) \tag{1}$$

2. Next, we want to use the values r_i of each asset (which we will call data points), for TDA analysis. In order to do so, we use a sliding window of length w (i.e. composed of w data points), and iterate over the whole array r, creating persistent diagrams on each iteration. In order to do so, we use the Vietoris-Rips solver provided by Rips(maxdim = 2) from the module ripser, which returns a persistence diagram using the function .fit_transform(r[a:b]), where a and b are the endpoints of the interval of data points of the cloud. Hence, on each iteration, the persistence diagrams of adjacent windows are obtained through the function .fit_transform(point cloud), and their Wasserstein distance is then measured using the function persim.wasserstein(first diagram, second diagram, matching=False) from the module persim. Notice that there are two free variables in the model: the length of the sliding window w, and the distance between adjacent windows d. The latter controls how much interaction there is between

each compared pair of persistence diagrams, while the former captures the amount of information used for computing each diagram. In a first test, we use $w = 25$ and $d = 22$, i.e. an overlap of only three points between adjacent diagrams, and almost a month of information for each window. The resulting Wasserstein distances for the case of HLT is shown in Figure 2, which clearly shows the prediction of the crash of the HLT stock in February-March 2020, corresponding to the beginning of the COVID. The prediction is marked by a very strong increase in Wasserstein distances beginning almost a month before the asset's crash.

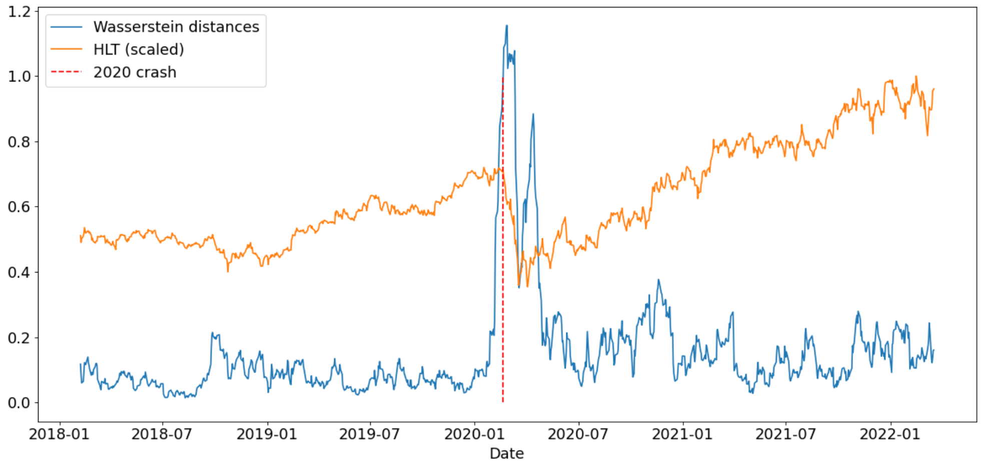

Fig. 1.— Prediction of the crash of Hilton stock in 2020 using the Wasserstein distances. w: 25; d: 22.

We now consider the case where we slightly vary the values of w and d. In the case $w = 30$ and $d = 5$ (high overlap between consecutive windows) we do not predict the crash of February 2020, see Figure 2. Hence, the values w and d have an impact on the prediction of crashes. The Model 2 will aim to quantify this impact and select the optimal pair (w, d) for each asset. In addition, if we take a closer loop to the previous two figures, one might actually realize that in the days previous to the COVID crash (i.e. when the W-distance highly increases), the W-distances are actually using information from the days of the crash and right after the crash, since each W-distance obtained for day i uses the window beginning at day i and finishing at day $i + w - 1$, and the window beginning at day $i + d$ and finishing at day $i + w + d - 1$. Therefore, if $w = 25$ and $d = 22$, each W-distance on day i is using information of up to 47 days after, which makes the prediction from Figures 2, 2 not actually *predictions*.

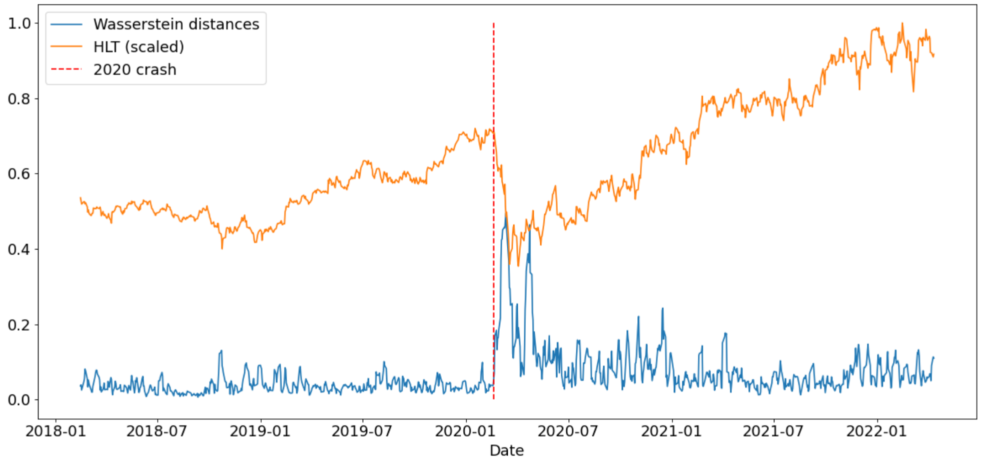

Fig. 2.— Prediction of the crash of Hilton stock in 2020 using the Wasserstein distances. w: 30; d: 5.

Therefore, we repeat the process, but this time cutting the information; in this new case the time series data finishes the day before 2020-02-20 (i.e. the day before the crash). Using the same algorithm, the result for $w = 25$ and $d = 22$ is shown in Figure 3. These results are far worse than the previous ones. Notice, however, that there is still a slight increase of the W-distances before the crash, so it may still be possible to make predictions. Indeed, our goal in the remaining of the paper will be to find an optimal way to make *real* predictions. Our novel approach is described next.

3. Model 2: Adaptative persistence homology

The two main free parameters of the PH model are w and d. As we have seen in the previous section, there are pairs of (w, d) that work better than others. How to chose then the best pair? There are 1580 pairs of these values if w goes from 20 to 59 and d from 1 to w. Thus, one should aim for an automatic way of finding the optimal (w, d), which is what we will do here. Our method works as follows:

1. Take an asset and its time series data from times a to b.

2. For each asset price value i (from $a + 1$ to $b - 1$), get the slope of the straight line fitting best the three points $i - 1$, $i1$, $i + 1$, in a least-squares sense. This slope is an approximation to the derivative of the function $P(i)$ (P for price) — which is in reality

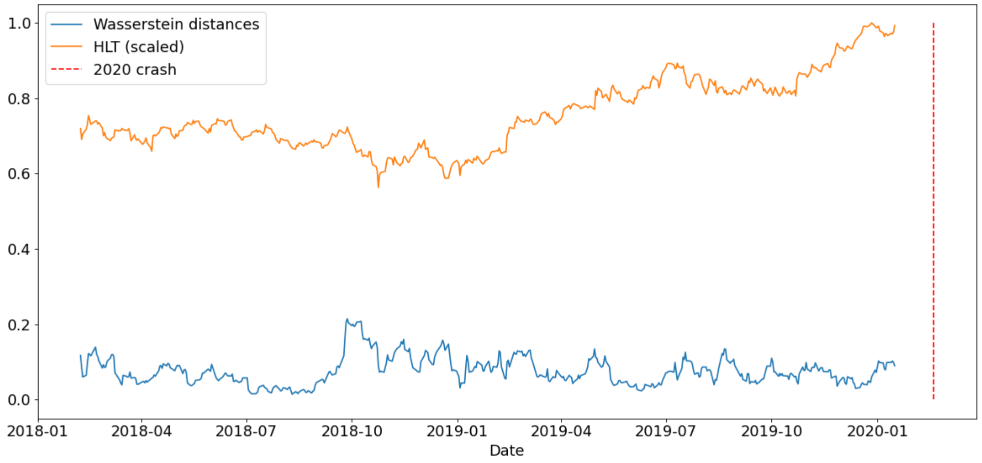

Fig. 3.— Attempt of prediction of the crash of Hilton stock in 2020 using the Wasserstein distances, with data only up to the day of the crash. w: 25; d: 22.

a discrete function — in the element i, i.e. day i.

3. Take all possible combinations of $(w,\ d)$ and compute the W-distances Wd_i in $[a,\ b]$. From each of these distances Wd_i, get also the slopes fitting the curve for every three points $i-1$, i, $i+1$, analogously to Step 2. Note that the slopes are measured because our goal is to predict the future slope of the curve $P(i)$, i.e. if the stock increases or decreases. More precisely, at time $i+w+d$, we want to know what the slope will of the price function will be the day after, i.e. at day $i+w+d+1$ (which is impossible to know at day $i+w+d$ since we would need to know $P(i+w+d+1)$ and $P(i+w+d+2)$). Hence, we only have the values of P up to day $i+w+d$ included. And the W-distances can only be computed up to day i (the second cloud used for Wd_i employs values of prices up to $P(i+w+d)$ included, so Wd_{i+1} would need values of the price we do not currently have). Therefore, we will use the slope of Wd_{i-1} as an indicator of the slope at time $i+w+d$ —or at least assume it is, and see how it actually performs.

4. Now that we have the slope of the r vector and of the W-distances on every day, we want to quantify how well the PH model is performing for predicting slopes. In order to do so, we create a value, ACC, representing the mean accuracy to predict rises or falls in the asset. More precisely, ACC is the frequency with which the model $(w,\ d)$ predicts the opposite slope of the price function —recall Figure 2; a good $(w,\ d)$ pair gives a W-distance function whose slope is opposite to the slope of the price function.

Given a pair (w, d), and noting the approximated slope as $\hat{S}$, ACC is calculated as:

$$ACC = \frac{1}{N}\sum_{i=1}^{N} Q(\hat{S}(Wd_i), \hat{S}(P(i+w+d+1)) \tag{2}$$

Where N is the number of Wd$_i$ values, and Q is given by:

$$Q(\hat{S}(Wd_i), \hat{S}(P(i+w+d+1))) = \begin{cases} 1 & \text{if } \hat{S}(Wd_i)\cdot\hat{S}(P(i+w+d+1) < 0 \\ 1 & \text{if } \hat{S}(Wd_i) = \hat{S}(P(i+w+d+1) = 0 \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

In other words, Q returns 1 if both slopes have opposite sign or are both equal to zero, and returns 0 otherwise. ACC is thus the frequency with which the W-distance function has predicted the opposite slope of the price function. Hence, a larger value of ACC (which can be only as large as 1) represents a good pair of (w, d). We then compute ACC for every pair $(w,d)_j$, obtaining a function $ACC(j)$ —where for simplicity we have ordered all pairs and represented with a single index j, going from 0 to 1579. All we have to do next is find the maximum of ACC and select the $(w,d)_j$ corresponding to that maximum. That pair will correspond to the optimal pair for predicting the slope of the function of prices.

4. Results

We run the algorithm described in Model 2, and as a return obtain the function $ACC(j)$, which is plotted in Figure 4. The maximum of ACC is obtained for (w, d)=(37, 36), with $ACC = 0.5727$, i.e. a prediction of the slope of price with ∼57% accuracy. With these two values, we then re-run Model 1 with the time cut right before the 2020 COVID crash. We would expect, if Model 2 was a good model, a clear increase of the slope of the W-distances around w+d+1 days before the crash. The resulting prediction with the optimal pair (w, d)=(37, 36) is shown in Figure 4. From the plot, it does not seem to predict the incoming crash. There is a slight increase of the W-values at the end, but it is not a significant marker. In addition, the last slope of the W-values (i.e. the most recent slope of the Wd$_i$'s that can be obtained) is -0.00991989. Hence, the last slope is not an indicator in this case of an incoming crash.

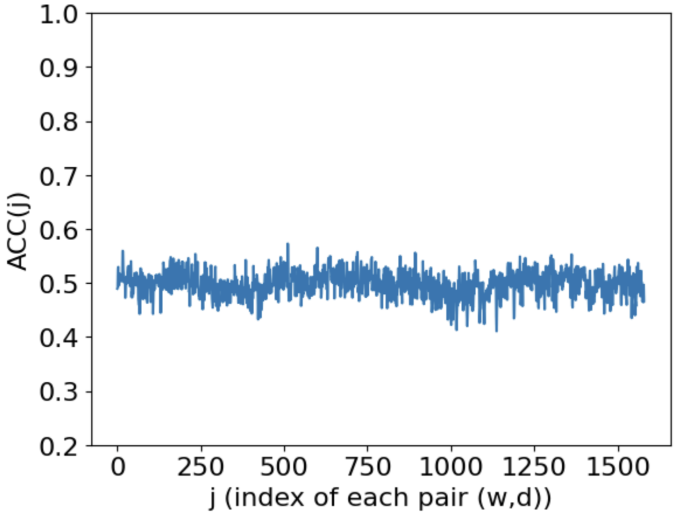

Fig. 4.— Function $ACC(j)$, the frequency of prediction of slope of the price function, with j being the index of each ordered pair (w, d).

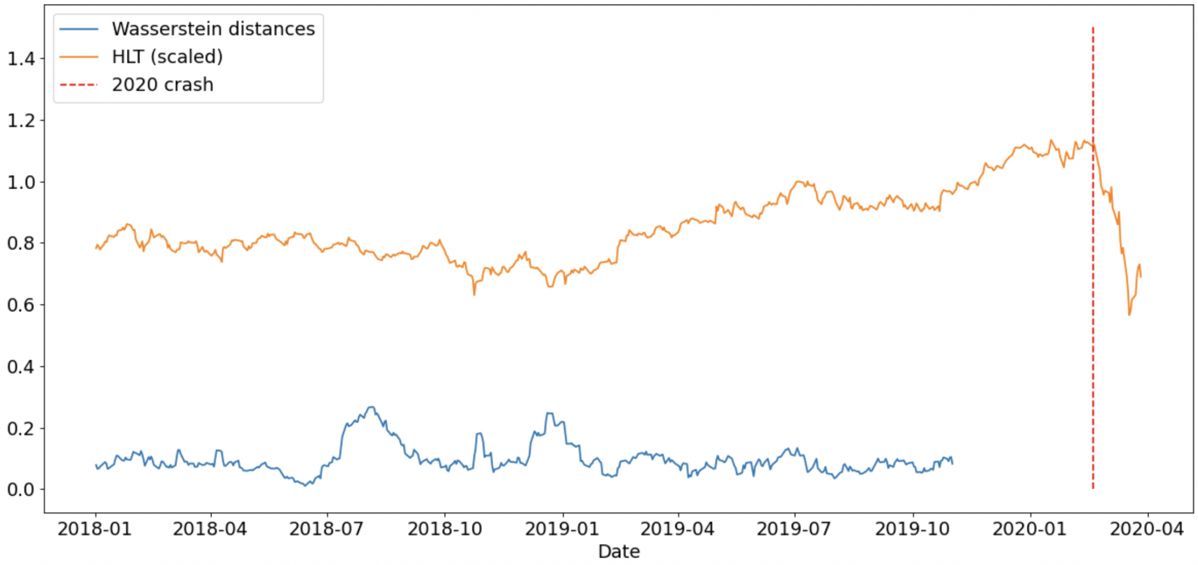

Fig. 5.— Temptative of prediction of the 2020 COVID crash with the optimal pair (w, d)=(37, 36) obtained from the adaptative model.

5. Conclusion

Our work has delved into topological data analysis and, more precisely, persistence homology. We have taken the work by Gidea et al. (5) and analyzed the prediction capabilities of persistence homology combined with Wasserstein distances for measuring differences in persistence diagrams. That approach has been defined as model 1. Although with model 1 variations in W-distances are related to crashes in asset values, we have found a flaw in the technique, which is that it uses data from the crash itself. That problem had not been assessed in the literature to our knowledge, and thus in order to solve the problem we have developped a new technique, adaptative persistence homology. The method finds for each different asset the optimal parameters of the persistence homology algorithm (specifically, size of the sliding window and distance between consecutive windows), for providing the most accurate estimation of the variations of the asset price over time. Our results, tested with the assets HLT, ˆFVX and AMZN, have yielded an optimal set of parameters (ω, d)=(37, 36), with an accuracy of slope sign prediction of 57.27%, against an average of slope sign prediction of 49.78% over all possible pairs. However, when testing the selected pair, the model does not seem to predict the crash. In addition, it is worth noting that Figure 4 suggests that the W-distances are not predicting the slopes, but rather only giving the correct sign of the slope randomly 50% of the time, with small variations due to noise or other unknown reasons. Hence, the choice of an optimal pair of parameters would not have a significant meaning; it would probably be a pair that had randomly performed better than the others. To conclude, our analysis of persistence homology proves that topological data analysis is not that accurate when predicting crashes once the data has been carefully selected. In addition, variations in Wasserstein distances do not seem to have an effect on variations in the stock market happening 20-60 days after. A solution could be using other distance functions, such as L^1 or L^2, or employing methods of topological data analysis other than persistence homology.

REFERENCES

Magnus Botnan and Michael Lesnick. Algebraic stability of zigzag persistence modules. *Algebraic & geometric topology*, 18(6):3133–3204, 2018.

H Edelsbrunner and JL Harer. Computational topology. an introduction, amer. math. soc. 2010.

Herbert Edelsbrunner. Persistent homology: theory and practice. 2013.

Herbert Edelsbrunner, John Harer, et al. Persistent homology-a survey. *Contemporary mathematics*, 453(26):257–282, 2008.

Marian Gidea and Yuri Katz. Topological data analysis of financial time series: Landscapes of crashes. *Physica A: Statistical Mechanics and its Applications*, 491:820–834, 2018.