Pairs Trading via Unsupervised Learning

Chulwoo Han* Durham University

Zhaodong He Durham University

Alenson Toh Nanyang Technological University

April 12, 2021

Abstract

This paper develops a pairs trading strategy via unsupervised learning. Unlike conventional pairs trading strategies that identify pairs based on return time series, we identify pairs by incorporating firm characteristics as well as price information. Firm characteristics are revealed to provide important information for pair identification and significantly improve the performance of the pairs trading strategy. Applied to the US stock market for the period from January 1980 to December 2020, the long-short portfolio constructed via the agglomerative clustering earns a statistically significant annualized mean return of 24% and a Sharpe ratio of 2.73. The strategy remains profitable after accounting for transaction costs and removing stocks below 20% NYSE-size quantile. A host of robustness tests confirm that the results are not driven by data snooping.

Keywords: Unsupervised learning; Pairs trading; k-means clustering; DBSCAN; Agglomerative clustering.
JEL Classification: G11, G12.

*Corresponding author.

1 Introduction

1.1 Motivation

Machine learning has gained momentum and been widely applied in various fields, including finance in recent years. It is estimated that 56% of hedge funds utilized machine learning in their investment decisions in 2018, as compared to only 20% in 2017 (Chahn, 2018). However, most of these applications are supervised learning-based, where they aim to predict returns, or reinforcement learning-based, which are used for selecting stocks for portfolio construction. In contrast, unsupervised learning is rarely used in the finance literature.

Unsupervised learning, also known as clustering, classifies a set of data into a group if they exhibit more significant similarity to one another, as compared to data that are classified into other groups. Different clustering methods can give different clustering results with the same dataset due to the difference in their clustering algorithms, which are discussed in greater detail in Section 3. Unsupervised learning has several applications in finance, such as industry classification and portfolio construction. In this paper, we explore the feasibility of unsupervised learning for pairs trading. Stocks with similar characteristics can be classified into a cluster via unsupervised learning. These stocks are likely to have similar future price movement, and one can find arbitrage opportunities by shorting overpriced stocks and longing underpriced stocks within the same cluster.

Existing pairs trading strategies typically identify pairs based on cointegration or some other time series related measurements (Gatev et al., 2006; Jacobs and Weber, 2015; Vidyamurthy, 2004; Rad et al., 2016; Huck, 2009, 2010). However, using only past price data can falsely identify spuriously correlated pairs that may not move in tandem in the future. Unsupervised learning allows us to identify similar stocks considering not only past price movement but also other firm characteristics. Stocks with similar characteristics are believed to co-move with a higher probability.

1.2 Objectives

This research aims to investigate the feasibility of unsupervised learning to find potential pairs of stocks to construct a long-short portfolio. We propose a framework to identify clusters from high dimensional financial data and implement three popular clustering algorithms to construct pair portfolios for statistical arbitrage. We choose k-means clustering, density-based spatial clustering of applications with noise (DBSCAN), and agglomerative clustering as representatives of each category of clustering methods.

The k-means clustering (MacQueen et al., 1967) is the pioneering and most popular clustering method, which assigns data points to various clusters by minimizing the within-cluster sum of squares (WCSS) between the centroids of the assigned clusters and the data points.

DBSCAN (Ester et al., 1996) is a density-based clustering method, where clusters are identified based on a continuous region of high density and are separated from each other by a continuous region of low density. Unlike k-means clustering which assigns all data points to clusters, the data

points in the low-density area are treated as outliers and are not assigned to any cluster.

The agglomerative clustering (Johnson, 1967) is one of the most representative hierarchical clustering methods. It is known as a "bottom-up" approach, where each data point is initially treated as a cluster on its own before being merged with other clusters as one moves up the hierarchy.

Our clustering-based pairs trading strategy consists of two steps. In the first step, stocks are clustered based on their past returns and characteristics. Stocks are grouped together if their prices move in tandem historically and have similar characteristics. We employ various firm characteristics as well as past returns as they are believed to be forward-looking and reduce the chance of falsely identifying spuriously correlated pairs.

The second step involves setting up a trading rule based on the divergence of the past one-month returns. Low (high) return stocks within the same cluster are deemed undervalued (overvalued), and a contrarian strategy is adopted by going long on the undervalued stocks and short on the overvalued stocks. More specifically, we open long and short positions if the return difference is greater than one cross-sectional standard deviation of the difference of past one-month returns. The long-short portfolio is rebalanced and reinvested at the end of every month.

Pairs trading reduces the market risk and makes the trading strategy profitable with a minimal market beta. In practice, investors do not trade only one pair but trade multiple pairs to form a well-diversified long-short portfolio. Our strategy allows the investor to choose a desired number of stocks by adjusting the threshold for the return divergence.

Applying clustering for pairs trading entails several difficulties and unexpected issues to be resolved. Firstly, clusters are unbalanced in their size. Some clusters have a sufficient number of stocks to form a portfolio, whereas other clusters have only a minimal number of stocks. In extreme cases, there are clusters that consist of only two stocks. DBSCAN, in particular, tends to yield a single colossal cluster comprised of the majority of the stocks, while the remaining stocks are identified as outliers, which suggests that the high-dimensional data structure has one big cluster that is exceptionally dense, and the rest of the stocks are scattered elsewhere.

Another issue is that some clustering methods, such as k-means clustering, assign all the stocks into clusters without identifying and excluding outliers. Hence, some of these clusters may contain dispersed outliers that are geometrically far from their assigned group centroids. Forming portfolios with such outliers will diminish profitability.

To address these issues, we test the clustering methods varying the clustering parameters and examine the impact of the number of clusters on the strategy performance. We also propose a method to remove outliers when using the k-means clustering.

1.3 Main Findings

We test our strategies on all listed stocks in the US stock market during the sample period of December 1979 to November 2020. The out-of-sample period is from January 1980 to December

2020 as we form a portfolio at the end of each month and hold it for one month.

Comparing the performance of the long-short portfolios constructed by the three clustering methods, we find that the agglomerative clustering significantly outperforms the others, attaining an annualised mean return of 23.8%, an annualized Sharpe ratio of 2.73, and a low maximum drawdown of 8.4%, which lasts only for one month. The k-means clustering and DBSCAN attain an annualised Sharpe ratio of 2.33 and 1.89, respectively, exhibiting great potential of unsupervised learning for pairs trading. They all outperform the S&P500 index and the conventional short-term reversal strategy significantly.

Our strategies also discover a significant alpha. The agglomerative clustering-based portfolio produces an economically and statistically significant monthly alphas of 1.8%, when regressed on the Fama-French three factors, Fama-French three factors plus momentum and short-term reversal factors, Fama-French five factors, and q^5 factors.

When stocks are clustered using only price information, the performance diminishes considerably; the Sharpe ratio decreases from 2.33 to 1.86 for k-means clustering, 1.89 to 1.57 for DBSCAN, and 2.73 to 1.71 for agglomerative clustering, and the differences are statistically significant. This result supports our hypothesis that the firm characteristics play a nontrivial role in identifying pairs.

We attempt to shed light on the black-box of the clustering methods and unveil sources of profitability. We find that all three clustering methods select stocks mainly from the manufacturing and financial sectors. This implies that the strategies are scalable as these are the sectors that contain the majority of the stocks in the market and their firms are usually large. This also implies that the firms in these sectors are more similar to each other within their respective sector and are more likely to move in tandem. Cluster are homogeneous in terms of industry, consisting of only one or few sectors. We also find that the long positions and short positions are often from different sectors, which implies that the profit does not only result from the divergence of the stocks in the same industry but also from the divergence between sectors.

The breakdown of the portfolios in size reveals that they choose more stocks from the upper half of the size deciles, which suggests that the profits are not driven by small-cap stocks.

Several robustness tests are conducted to ensure that the results are not sensitive to parameters and not driven by data snooping. The robustness check reveals that our models perform robustly in different parameter settings.

Excluding stocks below 20% NYSE-size quantile reduces the profitability of the strategies only slightly, suggesting that the profits are not driven by small-cap stocks and the strategies are highly scalable for asset management firms and hedge funds that typically trade large positions.

The turnover of the strategies is high at about 180%, which means half of the positions are replaced when the portfolio is rebalanced. This is mainly due to the nature of the contrarian strategy. Adding transaction costs lowers the Sharpe ratio, but the financial performance is still promising. The agglomerative strategy attains an annualized Sharpe ratio of 1.72 and a mean

return of 14.9% after subtracting 20 basis point transaction costs.

Our strategies perform exceptionally well during a financial crisis. All three strategies perform robustly and significantly outperform the benchmarks during the 2007 financial crisis and the 2020 market crash caused by the Covid-19 pandemic.

1.4 Contributions

The contribution of this paper is twofold. Firstly, to our knowledge, this is one of the earliest attempts to apply unsupervised learning to pairs trading, which extends the empirical asset pricing literature. We offer a framework to identify pairs via clustering methods and construct portfolios from them. Unsupervised learning appears to extract helpful information from high-noise financial data and identify meaningful pairs. The long-short portfolios derived from the clusters are more profitable and less volatile compared to those from conventional pairs trading strategies. This result suggests that stocks' behaviour and characteristics should be considered more thoroughly when selecting pairs for pairs trading, rather than relying on the statistical measures used in traditional pairs trading strategies.

Our study provides a big picture of how stock data in high-dimensional space are perceived through the eyes of the various clustering methods. The analysis of clustering characteristics shows that all clustering methods identify at least 60% of available stocks as outliers, implying more than half of the stocks are isolated and should not enter the pairs trading portfolios. For the remaining stocks, the k-means clustering and aggloermative clustering tend to form more than a hundred clusters of various sizes, whereas DBSCAN tends to form one enormous cluster that encompasses almost all the remaining stocks, and another tiny cluster containing less than ten stocks on average.

Secondly, our study reveals the importance of firm-level characteristics in identifying pairs. Unlike traditional pairs trading strategies, unsupervised learning can incorporate the information from firm characteristics as well as price-related information. In our study, it is found that adding firm characteristics reduces volatility and significantly improves the performance of the pairs trading strategies. To the best of our knowledge, this paper is the first attempt to incorporate firm characteristics in pairs trading.

This paper is organised as follows. Section 2 reviews existing pairs trading strategies in the literature as well as the financial applications of machine learning. Section 3 describes the data and feature generation process. It also details the working principle of the clustering methods and the trading algorithm. Section 4 presents the empirical results, including clustering behaviour, financial performance, factor regression analysis, and sub-period analysis. Section 5 performs various robustness checks, and Section 6 concludes.

2 Literature Review

This paper contributes to the literature on statistical arbitrage by proposing new ways to construct pairs trading strategies. More broadly, it extends the literature on financial applications of machine learning by employing unsupervised learning.

2.1 Existing Pairs Trading Strategies

Krauss (2017) classifies pairs trading strategies into four groups. The most mainstream approach is the cointegration approach, which involves the Engle and Granger (1987) cointegration test. Using this framework, Rad et al. (2016) implement a pairs trading strategy with time-varying trading costs. The strategy yields a mean monthly excess return of 33 basis points after transaction costs from 1962 to 2014. Intuitively, they discover that their pairs trading strategy is more profitable during a period of high volatility. Huck and Afawubo (2015) compare the cointegration method with the distance approach on the constituents of S&P500 and demonstrate that the cointegration method is stable and can generate profits consistently. They also address the meagre excess return of the distance approach after factoring in transaction costs. Clegg and Krauss (2018) extract pairs from the S&P500 constituents via partial cointegration. After including a mean-reverting component in the cointegration residuals, the portfolio yields an annualized return of more than 12% after transaction costs from 1990 to 2015.

The second approach is the distance approach, which identifies co-moving pairs based on a particular distance metric. Gatev et al. (2006) utilizes the Euclidean squared distance between the normalized prices of securities to identify pairs and form a long-short portfolio. Positions are opened when the distance of a pair is greater than two standard deviations of the historical distances between the pair. Positions are closed either when mean-reversion occurs or at the end of the trading period. This paper provides the main ideas and building blocks of pairs trading. Do and Faff (2010) extend the study of Gatev et al. (2006) to post 2008 financial crisis and discover that the strategy performs strongly during the recession period. The same study finds that pairs trading is more profitable for bank stocks. Jacobs and Weber (2015) extend the distance-based pairs trading strategy to 34 countries and discover that it is profitable in different markets, despite its varying performance with time.

The next approach to pairs trading is the time-series approach. Cummins and Bucca (2012) adopt a novel time-series model to identify pairs in the oil market, with a focus on West Texas Intermediate (WTI), Brent, heating oil, and gas oil from 2003 to 2010. Their pairs trading strategy yields a mean daily return between 0.07% and 0.55% and a Sharpe ratio greater than 2, prior to transaction costs. However, contrary to other pairs trading strategies that are usually more profitable during the 2008 financial crisis, the returns generated from this time-series model diminish during the recession. Elliott et al. (2005) propose another time-series approach for pairs trading based on the mean-reverting Gaussian Markov chain, but no backtesting is conducted on any

market.

The most novel and relatable approach to this study is to construct pairs via supervised learning. Huck (2009, 2010) implements an ensemble method that combines an Elman neural networks, a recurrent neural network, and ELECTRE III to identify pairs from the S&P500 constituent stocks.[1] When backtested from 1992 to 2006, the portfolio yields a weekly excess return of more than 0.8%. Krauss et al. (2017) convert pairs trading into a supervised learning problem by feeding several features into machine learning algorithms (deep neural networks, gradient-boosted-trees, and random forests). They forecast the probability of stock prices' upward moving and construct a daily rebalanced portfolio by entering long positions in the stocks with the highest probabilities and short positions in the stocks with the lowest probabilities. From 1992 to 2015, an equally-weighted portfolio that ensembles the three machine learning methods yields a daily excess return of 0.45% prior to transaction costs. Huck (2019) adds more than 600 predictors to his machine learning models hoping that the learning algorithms can digest a large dataset and extract useful information from the extremely noisy data. Besides lagged returns, the feature set consists of time information, size of stocks, indices, commodity prices, and many frequently used risk factors. However, the positive excess return is negated by transaction costs, which implies that adding too many features may not boost the performance of machine learning-based pairs trading. Kim and Kim (2019) apply deep reinforcement learning for pairs trading to dynamically determine the trading threshold and the stop-loss boundary, and report promising results in comparison to a static boundary.

There are other non-conventional methods to identify pairs for pairs trading strategies, which are less frequently used in the literature. Avellaneda and Lee (2010) utilize the principal component analysis (PCA) on the exchange-traded funds (ETFs) listed in the US stock market and attain an annualized Sharpe ratio of 1.51 from 2003 to 2007. Rad et al. (2016) construct pairs based on a Copula method and gain a monthly excess return of 5 basis points after transaction costs, while Xie et al. (2016) demonstrate that the Copula method outperforms the distance approach in terms of average excess returns generated.

2.2 Financial Applications of Unsupervised Learning

Financial applications of unsupervised learning are relatively rare. For the distance-based clustering method, Nanda et al. (2010) develop a portfolio strategy using k-means clustering, which is followed by the Markowitz model to create a portfolio. Fung et al. (2002) use a modified k-means clustering to cluster news articles, with the aim of analyzing the impact of news articles on stock prices. Ta and Liu (2016) adopt k-means clustering as one of their clustering methods to cluster stocks listed on the Ho Chih Minh Stock Exchange and study the impact of foreign ownership on Vietnamese stocks' volatility.

[1] ELECTRE III is an outranking method based on the concept of fuzzy logic.

Yang et al. (2014) adopt DBSCAN to identify suspicious financial transactions. DBSCAN is also one of the three unsupervised clustering methods explored by Pavlidis et al. (2006) to cluster time series data before applying artificial neural networks to forecast daily foreign exchange spot rates. Bini and Mathew (2016) also use DBSCAN as one of their clustering algorithms to cluster stock market data before applying regression models to forecast prices of the stocks listed on the National Stock Exchange of India.

Dose and Cincotti (2005) apply complete-link agglomerative clustering to the daily closing prices of the S&P500 constituents to create an index and an enhanced index tracking portfolio. Babu et al. (2012) apply agglomerative clustering on financial data before applying a recursive k-means clustering within the agglomerative clusters to predict the direction of stock prices.

2.3 Financial Applications of Supervised Learning

The volume of the literature on the financial application of machine learning has been blooming in recent years. Exceptional financial performances are reported when machine learning is applied to asset return prediction, which can be easily defined as a supervised learning problem. Using a classification tree, Moritz and Zimmermann (2016) develop a two-step portfolio sorting method and report significant financial performance. Messmer (2017) obtains significant excess returns by constructing a deep learning model with 68 firm characteristics as input features. Gu et al. (2020) conduct a comparative analysis of popular machine learning algorithms in the US stock market. They find that a deep neural network outperforms tree-based models and linear benchmarks, and the long-short portfolio obtained from the deep neural network achieves an annualized Sharpe ratio above 2.0. Han (2020) employs a deep neural network to predict the cross-section of stock returns in the US market and finds that the long-short portfolio strategy derived from the model can earn a Shape ratio greater than 2.5.

Machine learning is also found to perform well in other asset classes. Lim et al. (2019) employ a deep neural network for time-series momentum and show that it performs robustly in the futures market. Bianchi et al. (2020) apply the methodology of Gu et al. (2020) to the US Treasury bond market and find machine learning-based strategies are profitable also in the bond market.

3 Methodology

3.1 Data Description and Prepossessing

This section describes the procedure to generate and process the input features. The feature set consists of 48 momentum factors and 75 firm characteristics generated every month for the sample period from December 1979 to November 2020. The out-of-sample period is from January 1980 to December 2020.

3.1.1 Data Sample

The US equity market data available from the Center for Research in Security Prices (CRSP) are used for our empirical studies. All stocks with common shares (share code 10 or 11) listed on the New York Stock Exchange (NYSE), American Stock Exchange (AMEX), and Nasdaq (exchange code 1, 2, and 3) are included. During the sample period, stocks must have a valid price on the portfolio formation date (at the end of month t) and returns from $t-48$ to t. For stocks delisted from the exchanges, the delisted returns are computed using the method of Beaver et al. (2007): should the delisted return with dividend is available in the CRSP delist file, it is used; otherwise, the delisted return is assumed to be -30%.

3.1.2 Feature Generation

The feature set consists of two parts. The first set comprises 48 price momentums, *i.e.*, 1- to 48-month price momentums. The other set consists of 75 firm characteristics chosen from Green et al. (2017). Stocks are clustered using the features available at the end of month $t-1$ and the previous one-month return is used to identify undervalued and overvalued stocks.

Momentum features reflect the historical movement of stock prices and stocks with similar momentum features are expected to keep the similar pace in the future. Firm characteristics can reveal more information from accounting and asset pricing perspectives. Even when a pair of stocks have moved in tandem in the past, if they have distinct firm characteristics, they may diverge from each other in the future: we consider firm characteristics more forward-looking. This motivates us to use both the momentum features and firm characteristics for clustering. A pairs trading strategy typically uses only prices or returns and this is the first work that employs firm characteristics to identify pairs.

Momentum features The i-month price momentum at the end of month $t-1$ is defined as the cumulative return from month $t-i$ to $t-2$ for $i>1$ and as the previous one-month return for $i=1$:

$$\begin{aligned} mom_i &= r_{t-1}, \quad i=1, \\ mom_i &= \prod_{j=t-i}^{t-2}(r_j+1)-1, \quad i \in 2,...,48, \end{aligned} \tag{1}$$

where r_j denotes the return in month j.

Firm characteristics The 75 firm characteristics are chosen from the firm characteristics used in Green et al. (2017), excluding the variables that use data from IBES and other variables with many missing values. They are listed in Table A and their exact definitions can be found in Green

et al. (2017) or the references therein. The firm characteristics are generated using the SAS code in Jeremiah Green's website.[2]

3.1.3 Data Preprocessing

Scaling data and performing PCA prior to unsupervised clustering is common in various clustering applications (Siraj et al., 2009; Chen et al., 2013; Sarmento and Horta, 2020). We first normalize features using their cross-sectional means and standard deviations to eliminate any skewed clustering. Unsupervised clustering methods usually cluster data points based on a certain distance measure and features with higher magnitude, such as market capitalization, will have higher weightage in distance calculation without normalization. This would result in a skewed clustering, where stocks are clustered chiefly based on those features that have higher magnitude in nature.

Some features must contain more information than others for the identification of similar stocks, but the distance measures assign the same weight to all features. Therefore, inputting the features directly into a clustering algorithm can dilute the information contained in important features. By applying PCA to the features and using the principal components as the input, we can emphasize those important features when clustering the stocks. Another benefit of PCA is to alleviate the curse of dimensionality. Clustering algorithms are resource-intensive and reducing the feature dimension via PCA can save the computational cost significantly.[3]

The principal components entered into clustering are determined so that the principal components explain at least 99% of the total variation. On average, 84 principal components out of 123 are selected as input.

3.2 Clustering Methods

This section provides an in-depth discussion of the clustering algorithms we employ: k-means clustering, DBSCAN, and agglomerative clustering, which are representatives of partition-based, density-based, and hierarchical clustering, respectively. Clustering algorithms are unsupervised, which means they do not have the true values (labels) to target. Therefore, they do not require a large set of historical data to train and validate the model. By feeding the input data, clustering algorithms find meaningful groupings that cannot be conceived easily by the human brain.

Unlike supervised learning models, *e.g.*, deep learning or gradient boosting, which often require many hyperparameters, clustering algorithms require very few hyperparameters and there is little room for data snooping. We test the models with different hyperparameter values and assess their impact on the performance of the pairs trading strategy.

[2] https://sites.google.com/site/jeremiahrgreenacctg/home

[3] For example, the time complexity of k-means clustering for each iteration is $O(KNM)$, where K is the number of clusters, N is the number of samples, and M is the feature dimension.

3.2.1 k-Means Clustering

The k-means clustering (MacQueen et al., 1967) is perhaps the most popular clustering method. It requires the number of clusters, K, to be first specified. It then identifies K centroids and groups all data points to one of these clusters by minimizing the within-cluster sum of squares (WCSS) between data points and their respective centroids. The objective function is given by

$$W = WCSS = \sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik} \left\| x^i - \mu_k \right\|^2, \tag{2}$$

where x^i refers to i-th data point, μ_k the centroid of cluster k, $w_{ik} = 1$ if x^i belongs to cluster k, otherwise $w_{ik} = 0$, and N is the total number of data points. $\|\cdot\|$ denotes l_2 norm.

The k-means clustering is a two-step minimization problem. W is first minimized with respect to w_{ik} while keeping μ_k fixed to update the assignment of data points to clusters (Equation (3)). After that, W is minimized with respect to μ_k while keeping w_{ik} fixed to recompute the centroids (Equation (4)). The above steps are repeated until W is minimized.

$$w_{ik} = \begin{cases} 1 & \text{if } k = \text{argmin}_j \left\| x^i - \mu_j \right\|_2 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\begin{aligned} \frac{\partial W}{\partial \mu_k} &= 2 \sum_{i=1}^{N} w_{ik} \left(x^i - \mu_k \right) = 0 \\ \Rightarrow \mu_k &= \frac{\sum_{i=1}^{N} w_{ik} x^i}{\sum_{i=1}^{N} w_{ik}} \end{aligned} \tag{4}$$

As k-means clustering assigns all data points into the specified number of clusters, it does not have the capability to identify outliers. Stocks that are located far from others are still assigned to the nearest centroid. Hence, we adopt and modify the method used by Rahmah and Sitanggang (2016) and Elbatta (2012) to identify these stocks and remove them from the asset pool.

The procedure of removing outliers is as follows. For each data point, measure the distance to its respective centroid and also the distance to its nearest neighbour. Sort the distances to the nearest neighbour in ascending order and choose the distance at α percentile as the threshold, ϵ. Any stock whose distance to its centroid is greater than ϵ is regarded as an outlier and removed. The pseudo-code for this process is described in Algorithm 2 in the appendix. We set $\alpha = 0.5$, *i.e.*, identify a stock as an outlier when its distance to its centroid is greater than the median distance between nearest stocks.

For the number of clusters, $K = 5, 10, 50, 100, 500, 1000$, and 1500 are tested.

3.2.2 DBSCAN

The density-based spatial clustering of applications with noise (DBSCAN) identifies areas of high density in a high dimensional data space separated by regions of low-density (Ester et al., 1996). It

has two parameters; the minimum number of data points per cluster, $MinPts$, and the maximum distance between data points, ϵ, to be considered in the same cluster. Distance metrics such as Manhattan (l_1 norm), Euclidean (l_2 norm), or Minkowski distance can be used to measure the distance between data points.

DBSCAN first selects an arbitrary data point as a core point. Data points with at least $MinPts$ neighbours within ϵ are considered for the core point. All neighbouring points within ϵ from a core point are clustered together with the core point, in a process called direct density reachable. Any neighbours of these neighbouring points within ϵ are also included in the same cluster. This process is called density reachable. Non-core points in a cluster are called border points, and points in the same cluster are said to be density connected. Outliers or noises are points that are not density reachable from any of the core points and they are not assigned to any cluster (Schubert et al., 2017). The above process is repeated until all data points are visited. The pseudo-code of DBSCAN is described in Algorithm 3.

Hyperparameter Selection Following Aggarwal et al. (2001), who demonstrate that l_1 norm provides more significant and meaningful clustering results for high-dimensional datasets, we choose l_1 norm as the distance metric for DBSCAN. Following Birant and Kut (2007), $MinPts$ is set to be the natural logarithm of the total number of data points N. The minimum distance ϵ is set as the α percentile of the average distances to the nearest $MinPts$ neighbour points. This method is recommended by Ester et al. (1996), who proposed DBSCAN. Elbatta (2012) and Rahmah and Sitanggang (2016) also adopt the same method to determine ϵ. We use $\alpha = 0.1, \ldots, 0.9$ for the empirical analysis.

3.2.3 Agglomerative Clustering

Agglomerative clustering is a hierarchical clustering method and starts by treating individual data points as a cluster on its own before merging the clusters step by step until termination criteria are met (Gowda and Ravi, 1995). It requires the user to specify one of two hyperparameters: the number of clusters K or the maximum distance for clusters to be merged, ϵ, known as linkage distance.

Figure 1 illustrates the agglomerative clustering process. The six individual data points are initially treated as individual clusters. Their proximity to each other is calculated, and similar clusters are merged to form a new cluster (BC and DE), should their distance is less than ϵ. This merging process is repeated until there are no more clusters whose distance to another cluster is less than ϵ. If the distance between merged clusters is always below the specified ϵ, eventually all cluster are merged to form cluster ABCDEF. In general, the final clustering result consists of many small clusters rather than a few giant clusters that contain most data points.

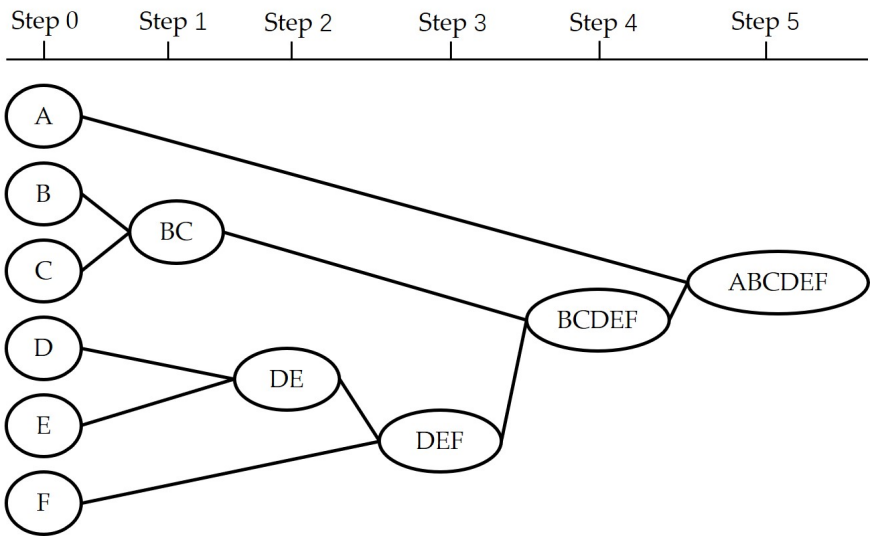


Figure 1: Agglomerative clustering

This figure presents a six-point dataset's hierarchical structure and shows how the agglomerative clustering algorithm groups data points from individual data points to the final cluster that includes all the data points.

Hyperparameter Selection In our empirical study, we specify the maximum distance ϵ rather than the number of clusters K, using a method similar to the method adopted for DBSCAN: ϵ is set as an α percentile of the distances between a pair of nearest data points. We implement agglomerative clustering with $\alpha = 0.1, \ldots, 0.9$. Similarly to DBSCAN, l_1 norm is used as the distance metric. For the linkage distance, *i.e.*, the distance between a pair of clusters, we adopt the average linkage, which is defined as the average distance between the data points in one cluster and the data points in another cluster. For instance, if there are two clusters G and H with N_G and N_H data points, respectively, the average linkage distance between G and H is defined as follows:

$$d_{\text{average}}(G, H) = \frac{1}{N_G \cdot N_H} \sum_{x^i \in G, x^j \in H} ||x^i - x^j||. \tag{5}$$

3.3 Portfolio Formation and Trading Strategy

At the end of each month during the sample period, we cluster the stocks using one of the clustering methods described in the previous section. Through clustering, stocks are either assigned to a cluster or identified as an outlier.

As we cluster stocks based on their price movements and firm characteristics up to month $t-1$, we expect the stocks in the same cluster that temporarily diverge from each other in month t to converge in the following month. Therefore, we detect pairs by examining the previous one-month return (mom_1). Stocks with a lower mom_1 are deemed undervalued and expected to rebound in the following month. Conversely, stocks with a higher mom_1 are deemed overvalued and expected to revert in the following month.

We sort stocks within each cluster on mom_1 and pair the stock with the highest mom_1 with the stock with the lowest mom_1, the second-highest mom_1 stock with the second-lowest mom_1 stock, and so forth. We then construct an equally-weighted long-short portfolio using the pairs whose mom_1 difference is greater than the cross-sectional standard deviation of all pairs' mom_1 differences. It is possible for some clusters to not have any pairs that meet the trading criterion if the mom_1 differences of their pairs are not wide enough. We hold the positions for one month and rebalance the portfolio at the end of each month. This cycle repeats until the end of the sample period.

The clustering-based long-short portfolios are compared with two benchmarks; the S&P500 index and the short-term reversal portfolio. We use the S&P500 index including dividends available from CRSP. For the short-term reversal portfolio, we sort stocks on mom_1 and go long on the stocks in the first decile and short on the stocks in the bottom decile.

4 Empirical results

This section presents the empirical results. We first summarize the clustering characteristics of each clustering method and then assess the financial performances of the pairs trading strategies from various perspectives. A host of robustness tests are conducted to ensure the findings are not driven by data snooping.

We test each clustering method with different parameter values and use the best performing case for the main results and later analyze the impact of the parameters on the strategy performance. For k-means clustering, the number of clusters $K = 5, 10, 50, 100, 500, 100,$ and 1500 are tested and $K = 500$ is chosen for the main results as it gives the highest Sharpe ratio. For DBSCAN and agglomerative clustering, $\alpha = 0.1, \ldots, 0.9$ are tested, and $\alpha = 0.1$ is chosen for DBSCAN and $\alpha = 0.3$ for agglomerative clustering for the same reason.

4.1 Clustering Characteristics

To give a big picture of how stocks are clustered, we analyze the clustering characteristics of the three clustering methods. The number of clusters vary significantly across the clustering methods. Table 1 and Figure 2 summarize the results.

The k-means clustering forms an average 130 clusters after removing outliers, with the range between 75 and 175.[4] The k-means clustering clusters average 1256 (39.8%) stocks every month and identifies the rest as outliers. Among the clustered stocks, average 64 (5.2%) stocks are contained in the largest cluster and 51 (4.1%) stocks are contained in the second largest cluster. Figure 2 shows that the number of clusters formed every month is consistent over the sample period. After

[4]The number of clusters, K, is set to 500, but the outlier removing process eliminates some clusters whose data points are far from each other and the remaining clusters are less than 200.

applying the one standard deviation threshold, average 309 stocks remain and are used to form a long-short portfolio.

In contrast to k-means clustering, DBSCAN tends to form one giant cluster and several small clusters, with an average of only two clusters every month. DBSCAN clusters only 377 (12.1%) stocks on average and 87.9% of the stocks are deemed as outliers. The giant cluster comprises almost all the clustered stocks (88.5%) and the remaining clustered stocks are assigned to one to six mini clusters. On average, DBSCAN trades 71 stocks every month.

Compared to k-means and DBSCAN, agglomerative clustering tends to form many small clusters (average 223), which comprises average 1030 (33.0%) stocks. Figure 2 reveals that there are no giant clusters as observed in DBSCAN and the number of clusters is rather consistent over the sample period. Agglomerative clustering trades average 171 (5.49%) stocks every month.

4.2 Strategy Performance

Table 2 reports the performances of the long-short portfolios constructed via the three clustering methods. The equally-weighted long-short portfolios outperform the S&P500 index and the benchmark reversal portfolio in terms of various financial performance indicators, revealing the clustering methods' capability to form meaningful clusters and generate significant profits.

Table 2 (a) reports the monthly return statistics of the clustering-based pairs trading strategies. It shows that the return of the long portfolio from all three clustering methods is economically and statistically significant with the t-statistic greater than 7. The return of the long-short portfolio, albeit smaller in magnitude, is even more significant with the t-statistic greater than 12. In particular, the agglomerative clustering yields the highest and most significant returns: 2.3% ($t = 8.89$) for the long portfolio and 2.0% ($t = 16.95$) for the long-short portfolio.

The mean returns of the long-short portfolios are greater than that of S&P500 and comparable to the mean return of the reversal portfolio. However, the long-short portfolios have significantly lower standard deviations.

Figure 3 presents the monthly return distributions in a box plot. It is apparent that all the clustering-based trading strategies have a narrower range of returns and exhibit smaller downside returns than the benchmarks, indicating that the clustering-based strategies are less volatile and less likely to experience drastic drops in returns.

Table 2 (b) reports annualized risk-return metrics. Five metrics are employed to assess the financial performance of the portfolios:

- Sharpe ratio: $\frac{Mean\ return}{standard\ deviation}$
- Sortino ratio: $\frac{Mean\ return}{Downside\ deviation}$
- Profit factor: $\frac{Gross\ profit}{Gross\ loss}$

- Maximum drawdown (MDD): the maximum observed loss from a peak to a trough over the portfolio formation period
- Calmar ratio: $\frac{Mean\ return}{MDD}$

The results suggest that the agglomerative clustering is the best performing clustering algorithm among the three clustering methods. The long-short portfolio has a high annualized mean excess return of 23.8% and a remarkably low standard deviation of 8.7%, resulting in the highest annualized Sharpe ratio of 2.73. Other performance metrics also favour agglomerative clustering: *e.g.*, it yields the lowest maximum drawdown of mere 8.4%. During the out-of-sample period of 41 years, there is no single year when the agglomerative clustering-based strategy suffers loss.

The long-short portfolio derived from k-means clustering also performs superbly, but has a slightly lower mean return (21.1%) and higher standard deviation (9.1%) compared to the one derived from agglomerative clustering, resulting in a Sharpe ratio of 2.33. Still, it significantly outperforms the benchmarks and suffers loss only in two years out of 41 years.

DBSCAN appears to be the worst-performing among the clustering methods. It has the highest mean return of 24.4%, but due to the relatively high standard deviation of 12.9%, it has the lowest Sharpe ratio. Other performance metrics also suggest that it is the least favourable method: *e.g.*, the maximum drawdown is 16.4%, significantly higher than those of k-means and agglomerative clustering. Nevertheless, it significantly outperforms the two benchmarks with an annualized Sharpe ratio of 1.89. An advantage of DBSCAN is that it requires on average only 71 stocks (35 pairs) to be traded every month, while the k-means and agglomerative clustering require more than 170 stocks to be traded every month.

Figure 4 shows the log-scale cumulative returns of the strategies over the out-of-sample period. The graph reaffirms that the pairs trading strategy via agglomerative clustering performs best with the highest cumulative return. DBSCAN performs remarkably well after the financial crisis in 2008 and yields a comparable cumulative return. The reversal portfolio performs exceptionally well before 2000, but its performance diminishes considerably thereafter. Remarkably, the portfolios constructed by the clustering methods perform even better during the financial crisis in 2008, when S&P500 and the reversal portfolio suffer a considerable loss.

4.3 Effects of Firm Characteristics

We hypothesize that the firm characteristics are forward-looking and would help clustering stocks and enhance the pairs trading strategies. To test the hypothesis, we repeat the same backtest using only the momentum features and compare the results with those reported earlier. Table 3 reports the results obtained from the clustering methods utilizing only the momentum features.

All the clustering methods perform worse after excluding the firm characteristics, in terms of the Sharpe ratio and other risk-adjusted performance metrics. Interestingly, they yield higher returns,

but the returns are more volatile resulting in lower Sharpe ratios. The mean returns of the long-short portfolios derived from k-means, DBSCAN, and agglomerative clustering are respectively 25.6%, 23.8%, and 25.0%, whereas they are respectively 21.1%, 24.4% and 23.8% when the firm characteristics are included. In contrast, the Sharpe ratios are reduced from 2.33 to 1.85 (k-means), 1.89 to 1.57 (DBSCAN), and 2.73 to 1.71 (agglomerative), and the differences are statistically significant with the t-statistic greater than 3 except for DBSCAN, whose t-statistic is 2.03. The maximum drawdown also increases substantially when the firm characteristics are removed: *e.g.*, the maximum drawdown of the agglomerative clustering increases from 8.4% to 32.6%.

The results above clearly suggest that the firm characteristics help identify similar stocks and enhance the performance of the pairs trading strategy. It is particularly notable that they reduce the volatility and downside risk substantially, making the strategy attractive to investors.

4.4 Factor Regression

To examine whether systematic risk factors can explain the long-short portfolios' returns, we employ four factor models and apply them to the equally-weighted portfolios constructed via agglomerative clustering. We focus on agglomerative clustering as it demonstrates the most promising performance among the clustering algorithms. The factor models considered are Fama and French (1996)'s three-factor model (FF3), FF3 plus the momentum and the short-term reversal factors (FF3+2), Fama and French (2015)'s five-factor model, and Hou et al. (2020)'s q^5 factor model. We include FF3+2 since the feature set includes momentum features and the trading strategy is a contrarian strategy, which closely relates to the short-term reversal factor. The Fama-French factors are downloaded from Kenneth French's website, and the q^5 factors are downloaded from the global-q website. For the q^5 factor model, the sample period ends in December 2019 as the site provides the factors only up to 2019. Table 4 reports the regression results.

The long-short portfolio constructed from agglomerative clustering yields an economically and statistically significant monthly alpha of 1.8% ($t = 15.29$) when regressed against FF3. The market beta is also positive and significant, suggesting that the trading strategy is dollar-neutral but not risk-neutral. The factor loading on HML is positive and weakly significant, implying that it favours firms with a higher book-to-market ratio. The adjusted R^2 from FF3 is 0.10.

The FF3+2 appears to have the best explanatory power for the strategy with the adjusted R^2 of 0.30. Nevertheless, the strategy generates a significant alpha of 1.9% ($t = 15.78$), which is the most significant among the four factor models. A positive, significant loading on the market factor is observed again. In addition, the momentum and the short-term reversal factors are negative and positive, respectively, and statistically significant. The positive loading on the short-term reversal factor is expected as our strategy is essentially a short-term reversal strategy with a carefully selected trading asset pool.

The FF5 and q^5 models have a similar explanatory power to FF3, and the alpha remains economically and statistically significant. The profitability and investment factors do not appear

to explain the returns of our strategy. The factors in q^5 are also unable to explain the returns, suggesting that the significant return of the strategy is not a result of excessive risk taking.

4.5 Size and Industry Breakdown of the Clusters

To investigate what drives the returns of the clustering-based pairs trading strategies, we categorize the stocks of the portfolios into size deciles and industry sectors. The results are summarized in Table 5, where the numbers are the average percentage of the stocks in each category.

Table 5 (a) reports the average proportion of the stocks in each size decile. Decile 1 represents the smallest stocks and decile 10 the largest stocks. All three clustering methods have a similar distribution of the stocks across size deciles and have the average size decile above 6.0. They select more than 10% of the stocks from each size decile above the median. This result corroborates the finding in Section 5 that the profits of the pairs trading strategies are not driven by small-cap stocks.

Table 5 (b) categorizes all stocks into eleven industry sectors using the North American Industry Classification System (NAICS).[5] The last column (Market) represents the proportions of all the stocks in the US market.

The k-means and agglomerative clustering have a similar distribution of the stocks across industries and select stocks mainly from the manufacturing sector, followed by the financial sector. Their industry composition is also similar to the industry composition of the entire market. DBSCAN, on the other hand, does not follow the distribution of the market but prefers stocks from the financial sector, followed by the manufacturing sector. Having many stocks from the financial sector is in line with Do and Faff (2010), who find that bank stocks are the most profitable when it comes to pairs trading.

The above results imply that the stocks in these industries are more similar to each other within their respective industry in terms of price movement and firm characteristics. Should any stock in these industries be overpriced or underpriced relative to the others in the same industry, its return is likely to revert back to the mean level offering a profit opportunity. Involving stocks mainly from these industries also explains why our clustering-based strategies remain profitable after excluding small-cap stocks: manufacturing and financial firms are usually large in market capitalization.

The number of industries within each cluster reveals that clusters are homogeneous in terms of industry, consisting of stocks from only one or few industries. The long-short portfolios constructed from k-means, DBSCAN, and agglomerative clustering respectively consist of 1.82, 3.45, and 1.42 industries, on average. The long and short portfolios individually consist of fewer industries: about 1.3 industries for k-means clustering, 2.7 for DBSCAN, and 1.1 for agglomerative clustering. This result implies that the profit does not only result from the divergence of the stocks in the same industry but also from the divergence between industries.

[5] https://www.naics.com/sic-codes-industry-drilldown/

4.6 Sub-period Analysis

We divide the sample period into five sub-periods and examine the performance of the clustering algorithms in each of these sub-periods. The results are reported in Table 6.

The first sub-period commences from January 1980 and ends in December 1999, prior to the widespread use of quantitative and machine learning algorithms in investment (Krauss et al., 2017). Even the benchmark reversal portfolio performs superbly during this period, attaining an annualized mean return of 34.5% and a Sharpe ratio of 2.08. However, the portfolios constructed via k-means and agglomerative clustering still outperform the benchmark significantly during this period: they attain an annualized mean returns of 23.7% and 25.8%, and the Sharpe ratios of 3.55 and 3.93, respectively. DBSCAN yields a Sharpe ratio of 2.05, which is comparable to that of the reversal portfolio, but it has a lower maximum drawdown and higher Sortino ratio. Overall, all three clustering methods perform superbly during this period.

The second sub-period ranges from January 2000 to December 2006, which coincides with the advent of quantitative and machine learning algorithms for investment (Vidyamurthy, 2004; Dempster and Leemans, 2006; Jae Won Lee, 2001). Cheaper and higher performing computing hardware has accelerated the adoption of quantitative approaches in this period, allowing more quantitative algorithms to be employed for portfolio construction. Hence, it is not surprising to observe a dip in the performance of all portfolios in this period compared to the previous period. Nevertheless, the clustering methods still manage to yield annualized mean returns above 25% and Sharpe ratios above 2.3. These values are significantly higher than those of the benchmark reversal portfolio, which are 21.6% and 0.51. DBSCAN performs particularly well in this period attaining a mean return of 33.9% and a Sharpe ratio of 2.79.

The third sub-period, January 2007 to December 2009, corresponds to the financial crisis caused by the collapse of housing prices in the US and its recovery thereafter. All three clustering methods remain profitable during the financial crisis and fare better than the benchmark reversal portfolio, except for DBSCAN, which performs comparably to the benchmark. To our surprise, agglomerative clustering performs superbly during the recession, with an annualized mean return of 23.5% and a Sharpe ratio of 2.11. Moreover, the maximum drawdown is mere 4.4%, which is significantly lower than those of the benchmarks; 55.1% for S&P500 and 21.2% for the reversal portfolio. This finding is in line with Do and Faff (2010) and Huck and Afawubo (2015), who find that pairs trading strategies perform better during a financial crisis.

The fourth sub-period ranges from January 2010 to December 2019. We discover that the returns generated by k-means and agglomerative clustering deteriorate during this period: they earn an annualized mean return below 15%. This finding is in line with Clegg and Krauss (2018), Rad et al. (2016), and Krauss et al. (2017), who observe that pairs trading strategies generate lower returns in recent years due to the widespread use of quantitative and machine learning algorithms. Any market inefficiency is rapidly arbitraged away eroding any potential profits to be generated. However, we still observe a significant amount of profits generated by DBSCAN in this period, with

an annualized mean return of 22.9% and a Sharpe ratio of 1.85. Agglomerative clustering attains a comparatively lower return of 14.5% and a Sharpe ratio of 1.78. The k-means clustering is found to be the worst-performing clustering method in this period, generating an annualized mean return of 11.8% and a Sharpe ratio of 1.47. Nevertheless, all three clustering methods outperform both benchmarks in terms of all metrics. The benchmark reversal portfolio underperforms S&P500 with a mean return of 3.7% and a Sharpe ratio of 0.23, suggesting that a naïve reversal strategy can no longer generate profits.

The last sub-period is year 2020, when the COVID-19 pandemic swept the world. To contain the spread of the virus, governments worldwide have implemented various degrees of lockdowns, which have severely depressed the global economy. Equity prices plummeted while the stock market volatility skyrocketed to an unprecedented level, which surpassed what was observed during the global financial crisis in 2008 as well as the Great Depression in 1929 (Baker et al., 2020).

The S&P500 index plunges by 27.5% until March 2020, while the reversal portfolio loses 15.9% in the same period. In contrast, the portfolios based on clustering fall by a much smaller magnitude. The agglomerative clustering loses only 3.5% and the other clustering methods also perform robustly with a maximum drawdown below 8%.

During the recovery from April 2020 onward, S&P500 recovers all the losses incurred till March 2020. The clustering-based strategies recover more rapidly and achieve significantly positive returns. DBSCAN performs particularly well generating a mean return higher than 50%. This result reaffirms the superiority and resilience of our strategies in a period of economic crisis.

5 Robustness check

Given the clustering methods' strong financial performance, coupled with the silent reproach of data snooping with machine learning algorithms, we perform a series of robustness check to ensure the performance is not sensitive to model parameters. Unlike many machine learning methods which often involve dozens of parameters and have high degrees of freedom, the clustering algorithms have few parameters, *e.g.*, the number of clusters K for the k-means clustering or the outlier threshold α for DBSCAN and agglomerative clustering. Besides the model parameters, we check the robustness against firm sizes, transaction costs, and the number of principal components.

5.1 Parameter Sensitivity of the Strategy

Table 7 analyzes the sensitivity of the pairs trading strategies to parameters. We find that the strategies are robust across a wide range of parameter values.

When we increase the number of clusters for k-means clustering from 5 to 1500, the number of clusters after removing outliers also increases, whereas the number of clustered stocks and the number of traded stocks decrease. This is because a large K generates many tiny clusters whose elements do not meet the trading criteria. The mean return increases with K as only a small

number of stocks that are closest to each other are paired and form the portfolio. But the Sharp ratio does not exhibit a monotonic trend and has its highest value when $K = 500$. It appears that the best performance requires a balance between selecting most profitable pairs and increasing diversification. Overall, the strategy is robust to K, with the mean return ranging between 17.5% and 27.4% and the Sharpe ratio ranging between 1.59 and 2.33.

DBSCAN clusters fewer stocks when α decreases, *i.e.* the maximum distance for a pair to be connected decreases. However, the number of clusters remains at 2. A smaller α results in a portfolio with fewer stocks and yields higher returns and Sharpe ratio, and the best performance occurs when $\alpha = 0.1$. The minimum α we test is 0.1, but decreasing α below 0.1 may improve the portfolio performance further. The strategy based on DBSCAN appears to be more sensitive than the one from k-means clustering: the mean return ranges between 12.7% and 24.4% and the Sharpe ratio ranges between 0.85 and 1.89.

Agglomerative clustering trades fewer stocks as the maximum distance for linkage decreases, *i.e.*, α decreases. Like in k-means clustering, selecting fewer stocks yields a higher return, but the Sharpe ratio starts to decrease when α is greater than 0.4. The performance varies rather widely, with the range of the mean return between 14.2% and 30.9% and the range of the Sharpe ratio between 1.22 and 2.73. Still, the strategy earns a mean return of 14.2% and a Sharpe ratio of 1.22 even in the worst case (α=0.9), in which only a small portion of the stocks (16.6%) are removed as outliers and the portfolio consists of almost 1000 stocks, on average.

Overall, all three clustering methods perform robustly within a reasonable range of the parameter values.

5.2 Size Effects

It might be that the returns generated by the clustering methods are attributed to illiquid, small-cap stocks. To assess the role of small-cap stocks, we repeat the empirical analysis after removing stocks with market capitalization smaller the 10% or 20% NYSE-size quantile. Table 8 (rows 'NYSE10' and 'NYSE20') reports the results.

Removing small-cap stocks lowers the mean return. However, it also reduces the volatility and the Sharpe ratio declines only moderately when the stocks below 10% NYSE-size quantile are excluded: the Sharpe ratio is decreased from 2.33 to 2.08 for k-means clustering, 1.89 to 1.69 for DBSCAN, and 2.73 to 2.38 for agglomerative clustering. Excluding the stocks below 20% NYSE-size quantile does not deteriorate the performance further. The mean return and the Sharpe ratio remain almost at the same level and even increase for k-means clustering.

Although small-cap stocks appear to have a certain extent of contribution to the strategies' profitability, the strategies can still generate significant profits when applied to relatively large stocks. This result reveals that the profits of the strategies are not a mere small-firm effect and they are scalable for large position sizes.

5.3 Value-Weighted Portfolio

For the pairs trading strategy to work as intended, the stocks in a pair should be equally weighted, *i.e.*, buy one stock and sell another for the same amount. As the pairs trading portfolio is simply a group of long-short pairs, it should also be equally weighted. Nevertheless, to further investigate the effects of firm size, we construct value-weighted pairs trading portfolios and report their performance in Table 8 (row 'VW'). For a fair comparison, the reversal portfolio is also value-weighted.

The value-weighted portfolios perform worse compared to their equally-weighted counterparts: the mean returns and Sharp ratios are approximately halved, which is in line with the findings of previous studies, *e.g.*, Gu et al. (2020); Han (2020). Nevertheless, they still outperform the value-weighted reversal portfolio and S&P500, generating higher returns and Sharpe ratios. In particular, the agglomerative clustering yields an annualized mean return of 14.2% and a Sharpe ratio of 1.02. In contrast, the S&P500 index and the reversal portfolio yield mean returns of 12.6% and 2.3% and Sharpe ratios of 0.84 and 0.09, respectively.

5.4 Impact of Transaction Costs

Our strategies are essentially a contrarian strategy and have a relatively high turnover (about 180%). But they require rebalancing only once a month and transaction costs should have limited impact. Table 8 (row '10 bps' and '20 bps') presents the portfolios' risk-return metrics after taking transaction costs into account. The transaction cost is assumed to be either 10 or 20 basis points (bp) per side of transaction.

After factoring in 10 bp transaction costs, the portfolio constructed via agglomerative clustering earns an annualized mean return of 19.4% and a Sharpe ratio of 2.23. DBSCAN and k-means clustering earn annualized mean returns of 16.8% and 20.0% and annualized Sharpe ratios of 1.86 and 1.56, respectively, which are still promising and higher than those of the benchmarks. Even with the conservatively assumed transaction cost of 20 bps, agglomerative clustering performs impressively attaining an annualized mean return of 19.4% and a Sharpe ratio of 1.72. The k-means clustering and DBSCAN earn annualized mean returns of 12.4% and 15.5% and Sharpe ratios of 1.37 and 1.21, respectively, and outperform the benchmarks. These results suggest that our strategies are likely to remain profitable even when the transaction costs are relatively high.

5.5 Trading Threshold

Previously, we use one standard deviation of the past one-month return difference as a threshold to trigger pairs trading. Here, we test different thresholds and examine their impacts on portfolio performance. A larger threshold selects pairs that have diverged more significantly in the previous month and trades fewer stocks. Table 8 (row '0.5 std', '1.5 std', '10%', and '20%') reports the results from different thresholds.

When the threshold is set to be 0.5 times the standard deviation (row '0.5 std'), the stocks

traded are doubled, and the returns become lower while the Sharpe ratios change little. Raising the threshold to 1.5 times the standard deviation (row '1.5 std') increases the return, but lowers the Sharpe ratio, which can be attributed to the reduced diversification effect. Still, our strategies outperform the benchmarks in both cases.

We also test a crude approach that simply chooses a certain percentile of the stocks from the top and the bottom. When the portfolio is formed by going long on the 10% lowest return stocks and short on the 10% highest return stocks within each cluster (row '10%'), it obtains a high Sharpe ratio, but the return tends to get lower except for DBSCAN, which earns a similar level of return: the Sharpe ratios of k-means, DBSCAN, and agglomerative clustering are respectively 2.32, 1.86, and 2.52, and the mean returns are 16.8%, 23.2%, and 12.0%. Choosing more extreme return stocks with 5% threshold (row '5%') renders similar results.

Choosing only a few pairs that have diverged significantly can yield a higher return, but the Sharpe ratio can deteriorate if the portfolio is not well diversified. A balance between the return and the volatility needs to be found. The result above suggests that the strategies perform robustly for a range of the threshold. It also suggests that determining the threshold dynamically based on the cross-sectional dispersion of the previous one-month return performs better than selecting a fixed proportion of the stocks.

5.6 Number of Principal Components

Table 8 (row 'PCA95' and 'PCA100') reports the sensitivity of the clustering methods to the number of principal components. We choose the number of principal components so that they can explain 95% or 100% of the variation. We use 100% since the objective of the PCA in our application is not to reduce the dimension, but to transform the feature space so that the principal components rather than the raw features receive the same weight in the distance metric. On average, 62 out of 123 principal components are selected at 95% and 84 are selected at 99%.

The results suggest that the pairs trading strategies are robust to the change of the number of principal components. Most risk-return metrics change only slightly with the number of principal components. Clustering algorithms are notoriously slow for a large set of high-dimensional data, and choosing the principal components at a lower variance level, *e.g.*, 95%, may save the computational costs significantly without sacrificing portfolio performance.

6 Conclusion

In this paper, we develop pairs trading strategies via unsupervised learning. We employ three clustering methods, *i.e.*, k-means clustering, DBSCAN, and agglomerative clustering, and utilize the information from firm characteristics as well as past returns.

We test our strategies on the US stock market during the period of January 1980 to December 2020 and find all three strategies significantly outperform the market and the short-term reversal

strategy. The agglomerative clustering-based strategy appear to be the best performing strategy with an annualised mean excess return of 23.8% and an annualized Sharpe ratio of 2.73. It also has a low maximum drawdown of 8.4%, which lasts only for one month. Factor models cannot explain the significant profits of the strategies. The strategies perform robustly during financial crises, such as the 2007 financial crisis and the 2020 market crash.

Firm characteristics are revealed to be an important source of information in identifying pairs. Clustering stocks based on their characteristics as well as past returns significantly reduces the volatility and downside risk of the strategies and improves performance.

The clustering methods select stocks mainly from the manufacturing and financial sectors and from mid- to large-cap deciles, making the strategies highly scalable. Each cluster consists of only one or few sectors, which implies that stocks in the same industry tend to behave in a similar manner. It appears that the profit of the strategies does not only result from the divergence of the stocks in the same industry but also from the divergence between industries.

A sub-period analysis and robustness tests reveal that the strategies perform robustly under different conditions and are not sensitive to changes of model parameter values.

In conclusion, clustering stocks incorporating the information from firm characteristics improves the identification of stock pairs and the performance of the pairs trading strategy.

References

Aggarwal, C.C., Hinneburg, A., Keim, D.A., 2001. On the surprising behavior of distance metrics in high dimensional space, in: International conference on database theory, Springer. pp. 420–434.

Avellaneda, M., Lee, J.H., 2010. Statistical arbitrage in the us equities market. Quantitative Finance 10, 761–782.

Babu, M.S., Geethanjali, N., Satyanarayana, B., 2012. Clustering approach to stock market prediction. International Journal of Advanced Networking and Applications 3, 1281.

Baker, S.R., Bloom, N., Davis, S.J., Kost, K.J., Sammon, M.C., Viratyosin, T., 2020. The unprecedented stock market impact of COVID-19. Technical Report. National Bureau of Economic Research.

Beaver, W., McNichols, M., Price, R., 2007. Delisting returns and their effect on accounting-based market anomalies. Journal of Accounting and Economics 43, 341–368.

Bianchi, D., Büchner, M., Tamoni, A., 2020. Bond Risk Premiums with Machine Learning. The Review of Financial Studies .

Bini, B., Mathew, T., 2016. Clustering and regression techniques for stock prediction. Procedia Technology 24, 1248–1255.

Birant, D., Kut, A., 2007. St-dbscan: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering 60, 208–221.

Chahn, 2018. Barclayhedge survey: Majority of hedge fund pros use AI/machine learning in investment strategies. Technical Note .

Chen, S., Lach, J., Amft, O., Altini, M., Penders, J., 2013. Unsupervised activity clustering to estimate energy expenditure with a single body sensor, in: 2013 IEEE international conference on body sensor networks, IEEE. pp. 1–6.

Clegg, M., Krauss, C., 2018. Pairs trading with partial cointegration. Quantitative Finance 18, 121–138.

Cummins, M., Bucca, A., 2012. Quantitative spread trading on crude oil and refined products markets. Quantitative Finance 12, 1857–1875.

Dempster, M.A., Leemans, V., 2006. An automated fx trading system using adaptive reinforcement learning. Expert Systems with Applications 30, 543–552.

Do, B., Faff, R., 2010. Does simple pairs trading still work? Financial Analysts Journal 66, 83–95.

Dose, C., Cincotti, S., 2005. Clustering of financial time series with application to index and enhanced index tracking portfolio. Physica A: Statistical Mechanics and its Applications 355, 145–151.

Elbatta, M.N., 2012. An improvement for DBSCAN algorithm for best results in varied densities. Ph.D. thesis. The Islamic University-Gaza.

Elliott, R.J., Van Der Hoek, J., Malcolm, W.P., 2005. Pairs trading. Quantitative Finance 5, 271–276.

Engle, R.F., Granger, C.W., 1987. Co-integration and error correction: representation, estimation, and testing. Econometrica , 251–276.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press. p. 226–231.

Fama, E.F., French, K.R., 1996. Multifactor explanations of asset pricing anomalies. The Journal of Finance 51, 55–84.

Fama, E.F., French, K.R., 2015. A five-factor asset pricing model. Journal of Financial Economics 116, 1–22.

Fung, G.P.C., Yu, J.X., Lam, W., 2002. News sensitive stock trend prediction, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer. pp. 481–493.

Gatev, E., Goetzmann, W.N., Rouwenhorst, K.G., 2006. Pairs trading: Performance of a relative-value arbitrage rule. The Review of Financial Studies 19, 797–827.

Gowda, K.C., Ravi, T., 1995. Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity. Pattern recognition 28, 1277–1282.

Green, J., Hand, J.R.M., Zhang, X.F., 2017. The characteristics that provide independent information about average U.S. monthly stock returns. The Review of Financial Studies 30, 4389–4436.

Gu, S., Kelly, B., Xiu, D., 2020. Empirical asset pricing via machine learning. The Review of Financial Studies 33(5), 2223–2273.

Han, C., 2020. Bimodal characteristic returns and predictability enhancement via machine learning. Working Paper .

Hou, K., Mo, H., Xue, C., Zhang, L., 2020. An augmented q-factor model with expected growth. Review of Finance .

Huck, N., 2009. Pairs selection and outranking: An application to the S&P 100 index. European Journal of Operational Research 196, 819–825.

Huck, N., 2010. Pairs trading and outranking: The multi-step-ahead forecasting case. European Journal of Operational Research 207, 1702–1716.

Huck, N., 2019. Large data sets and machine learning: Applications to statistical arbitrage. European Journal of Operational Research 278, 330–342.

Huck, N., Afawubo, K., 2015. Pairs trading and selection methods: is cointegration superior? Applied Economics 47, 599–613.

Jacobs, H., Weber, M., 2015. On the determinants of pairs trading profitability. Journal of Financial Markets 23, 75–97.

Jae Won Lee, 2001. Stock price prediction using reinforcement learning, in: ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570), pp. 690–695 vol.1.

Johnson, S.C., 1967. Hierarchical clustering schemes. Psychometrika 32, 241–254.

Kim, T., Kim, H.Y., 2019. Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries. Complexity 2019.

Krauss, C., 2017. Statistical arbitrage pairs trading strategies: Review and outlook. Journal of Economic Surveys 31, 513–545.

Krauss, C., Do, X.A., Huck, N., 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research 259, 689–702.

Ledoit, O., Wolf, M., 2008. Robust performance hypothesis testing with the sharpe ratio. Journal of Empirical Finance 15, 850–859.

Lim, B., Zohren, S., Roberts, S., 2019. Enhancing time-series momentum strategies using deep neural networks. The Journal of Financial Data Science 1, 19–38.

MacQueen, J., et al., 1967. Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, CA, USA. pp. 281–297.

Messmer, M., 2017. Deep learning and the cross-section of expected returns. Econometric Modeling: Capital Markets - Asset Pricing eJournal .

Moritz, B., Zimmermann, T., 2016. Tree-based conditional portfolio sorts: The relation between past and future stock returns. Working Paper .

Nanda, S., Mahanty, B., Tiwari, M., 2010. Clustering indian stock market data for portfolio management. Expert Systems with Applications 37, 8793–8798.

Pavlidis, N.G., Plagianakos, V.P., Tasoulis, D.K., Vrahatis, M.N., 2006. Financial forecasting through unsupervised clustering and neural networks. Operational Research 6, 103–127.

Rad, H., Low, R.K.Y., Faff, R., 2016. The profitability of pairs trading strategies: distance, cointegration and copula methods. Quantitative Finance 16, 1541–1558.

Rahmah, N., Sitanggang, I.S., 2016. Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra, in: IOP Conference Series: Earth and Environmental Science, IOP Publishing. p. 012012.

Sarmento, S.M., Horta, N., 2020. Enhancing a pairs trading strategy with the application of machine learning. Expert Systems with Applications 158, 113490.

Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X., 2017. Dbscan revisited, revisited: why and how you should (still) use dbscan. ACM Transactions on Database Systems (TODS) 42, 1–21.

Siraj, M.M., Maarof, M.A., Hashim, S.Z.M., 2009. Intelligent clustering with pca and unsupervised learning algorithm in intrusion alert correlation, in: 2009 Fifth International Conference on Information Assurance and Security, IEEE. pp. 679–682.

Ta, V.D., Liu, C.M., 2016. Stock market analysis using clustering techniques: the impact of foreign ownership on stock volatility in vietnam, in: Proceedings of the Seventh Symposium on Information and Communication Technology, pp. 99–106.

Vidyamurthy, G., 2004. Pairs Trading: quantitative methods and analysis. volume 217. John Wiley & Sons.

Xie, W., Liew, R.Q., Wu, Y., Zou, X., 2016. Pairs trading with copulas. The Journal of Trading 11, 41–52.

Yang, Y., Lian, B., Li, L., Chen, C., Li, P., 2014. Dbscan clustering algorithm applied to identify suspicious financial transactions, in: 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, IEEE. pp. 60–65.

Table 1: Clustering characteristics

This table reports the clustering results from the k-means clustering, DBSCAN, and agglomerative clustering. Stocks are clustered every month during the test period from 1990.01 to 2020.12, and the reported values are time-series averages. The values in parentheses are the percentages against the total number of stocks.

	k-means	DBSCAN	Agglomerative
Number of clusters	130	2	223
Number of stocks in total	3157	3157	3157
Number of stocks in clusters	1256 (39.76)	377 (12.07)	1030 (32.96)
Number of outliers	1901 (60.24)	2780 (87.93)	2127 (67.04)
Number of stocks in the biggest cluster	64 (5.21)	336 (88.52)	150 (14.32)
Number of stocks in the second biggest cluster	51 (4.1)	33 (9.22)	74 (7.1)
Number of stocks in the third biggest cluster	45 (3.63)	6 (1.83)	50 (4.72)
Number of stocks traded	309 (9.66)	71 (2.34)	171 (5.49)

Table 2: Strategy performance

This table reports the performance of the equally-weighted pairs trading portfolios constructed via k-means, DBSCAN, and agglomerative clustering. The test period is from 1980.01 to 2020.12. Panel (a) reports the summary statistics of the monthly returns, and panel (b) reports annualized risk-return metrics. The t-statistic in panel (a) is the Newey–West t-statistic. The t-statistic in panel (b) tests the difference of the Sharpe ratio from the Sharpe ratio of the benchmark reversal portfolio. It is computed using the method of Ledoit and Wolf (2008). S&P500 denotes the S&P500 index with dividends and Reversal denotes the equally-weighted short-term reversal portfolio.

(a) Monthly return summary statistics

	k-means			DBSCAN			Agglomerative			Benchmarks	
	L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
Mean	0.023	0.005	0.018	0.021	0.001	0.020	0.023	0.003	0.020	0.012	0.020
Standard deviation	0.059	0.047	0.026	0.054	0.047	0.037	0.055	0.046	0.025	0.049	0.070
Standard error	0.003	0.002	0.001	0.002	0.002	0.002	0.003	0.002	0.001	0.002	0.003
t-statistic	7.965	2.222	14.006	8.443	0.326	12.007	8.886	1.409	16.953	5.185	6.362
Min	-0.308	-0.266	-0.080	-0.312	-0.464	-0.123	-0.262	-0.348	-0.058	-0.256	-0.363
25%	-0.006	-0.019	0.002	-0.007	-0.021	-0.001	-0.005	-0.019	0.004	-0.015	-0.011
50%	0.025	0.010	0.015	0.022	0.005	0.017	0.024	0.007	0.018	0.014	0.014
75%	0.051	0.032	0.031	0.052	0.025	0.038	0.053	0.030	0.032	0.041	0.042
Max	0.310	0.202	0.159	0.216	0.156	0.180	0.262	0.167	0.158	0.185	0.800
Skew	-0.281	-0.955	1.117	-0.708	-2.347	0.681	-0.409	-1.490	1.077	-0.621	3.135
Kurtosis	4.645	4.642	4.250	4.943	20.624	2.215	3.738	9.106	3.428	3.194	34.244

(b) Annualized risk-return metrics

	k-means			DBSCAN			Agglomerative			Benchmarks	
	L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
Mean return	0.272	0.060	0.211	0.253	0.009	0.244	0.276	0.038	0.238	0.143	0.239
Standard deviation	0.206	0.164	0.091	0.185	0.163	0.129	0.190	0.160	0.087	0.171	0.243
Sharpe ratio	1.319	0.367	2.331	1.365	0.054	1.891	1.454	0.234	2.729	0.840	0.983
t-statistic			8.509			4.602			9.879		
Downside deviation	0.159	0.136	0.042	0.150	0.151	0.068	0.147	0.140	0.035	0.131	0.141
Sortino ratio	1.708	0.445	4.994	1.690	0.058	3.567	1.876	0.268	6.819	1.096	1.693
Gross profit	17.036	9.889	9.989	15.781	7.985	12.632	16.582	8.857	10.805	12.316	15.548
Gross loss	-5.896	-7.418	-1.320	-5.411	-7.623	-2.624	-5.281	-7.319	-1.042	-6.432	-5.755
Profit factor	2.889	1.333	7.569	2.917	1.048	4.814	3.140	1.210	10.374	1.915	2.702
Profitable years	36	28	39	37	23	40	37	26	41	36	33
Unprofitable years	5	13	2	4	18	1	4	15	0	5	8
Maximum drawdown	-0.480	-0.579	-0.112	-0.522	-0.816	-0.164	-0.492	-0.603	-0.084	-0.551	-0.503
Calmar ratio	0.566	0.104	1.885	0.484	0.011	1.488	0.561	0.062	2.836	0.261	0.475
Turnover	0.906	0.924	1.830	0.931	0.952	1.883	0.933	0.946	1.879		

Table 3: Strategy performance without firm characteristics

This table reports the performance of the equally-weighted pairs trading portfolios constructed via k-means, DBSCAN, and agglomerative clustering utilizing only momentum features. The test period is from 1980.01 to 2020.12. Panel (a) reports the summary statistics of the monthly returns, and panel (b) reports annualized risk-return metrics. The t-statistic in panel (a) is the Newey–West t-statistic. The t-statistic in panel (b) tests the difference of the Sharpe ratio from the corresponding Sharpe ratio in Table 2. It is computed using the method of Ledoit and Wolf (2008).

(a) Monthly return summary statistics

	k-means			DBSCAN			Agglomerative			Benchmarks	
	L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
Mean	0.025	0.003	0.021	0.024	0.004	0.020	0.024	0.003	0.021	0.012	0.020
Standard deviation	0.075	0.053	0.040	0.070	0.054	0.044	0.077	0.057	0.042	0.049	0.070
Standard error	0.004	0.003	0.002	0.003	0.003	0.002	0.004	0.003	0.002	0.002	0.003
t-statistic	6.876	1.285	11.093	7.297	1.688	9.454	6.466	1.263	10.140	5.185	6.362
Min	-0.373	-0.274	-0.137	-0.369	-0.464	-0.083	-0.296	-0.260	-0.182	-0.256	-0.363
25%	-0.014	-0.023	-0.001	-0.004	-0.015	-0.003	-0.015	-0.025	-0.001	-0.015	-0.011
50%	0.020	0.006	0.016	0.021	0.006	0.014	0.021	0.006	0.016	0.014	0.014
75%	0.056	0.030	0.036	0.047	0.025	0.037	0.055	0.031	0.035	0.041	0.042
Max	0.480	0.322	0.303	0.615	0.388	0.395	0.497	0.379	0.266	0.185	0.800
Skew	0.780	-0.109	1.516	1.840	-0.348	2.259	1.158	0.450	1.212	-0.621	3.135
Kurtosis	7.390	5.595	7.994	17.883	20.097	13.615	7.319	6.598	6.862	3.194	34.244

(b) Annualized risk-return metrics

	k-means			DBSCAN			Agglomerative			Benchmarks	
	L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
Mean return	0.295	0.039	0.256	0.291	0.053	0.238	0.291	0.041	0.250	0.143	0.239
Standard deviation	0.259	0.183	0.138	0.244	0.188	0.152	0.268	0.198	0.146	0.171	0.243
Sharpe ratio	1.140	0.215	1.858	1.195	0.281	1.571	1.083	0.207	1.709	0.840	0.983
t-statistic	2.552	2.503	3.878	1.608	2.292	2.030	3.496	0.250	5.009		
Downside deviation	0.171	0.138	0.070	0.171	0.162	0.066	0.163	0.141	0.088	0.131	0.141
Sortino ratio	1.730	0.286	3.630	1.703	0.327	3.591	1.781	0.290	2.853	1.096	1.693
Gross profit	19.255	9.925	12.790	17.544	9.051	12.733	19.460	10.483	12.977	12.316	15.548
Gross loss	-7.154	-8.312	-2.302	-5.603	-6.882	-2.960	-7.547	-8.808	-2.740	-6.432	-5.755
Profit factor	2.691	1.194	5.555	3.131	1.315	4.302	2.579	1.190	4.737	1.915	2.702
Profitable years	36	26	39	37	27	38	34	24	38	36	33
Unprofitable years	5	15	2	4	14	3	7	17	3	5	8
Maximum drawdown	-0.522	-0.656	-0.137	-0.477	-0.624	-0.129	-0.514	-0.656	-0.326	-0.551	-0.503
Calmar ratio	0.566	0.060	1.864	0.611	0.085	1.846	0.566	0.062	0.767	0.261	0.475
Turnover	0.857	0.933	1.790	0.932	0.981	1.913	0.884	0.954	1.838		

Table 4: Factor regression

This table reports the factor regression results of the equally-weighted pairs trading portfolio constructed via agglomerative clustering. FF3, FF3+2, FF5, and q^5 respectively denote the Fama-French three factors, FF3 plus momentum and short-term reversal factors, Fama-French five factors, and Hou-Mo-Xue-Zhang q^5 factors. The sample period is from 1980.01 to 2020.12 except for q^5, for which the period ends in 2019.12. The values in parentheses are the Newey-West t-statistics.

	FF3	FF3+2	FF5	q^5
Intercept	0.018***	0.019***	0.018***	0.019***
	(15.294)	(15.776)	(14.854)	(13.884)
Market	0.159***	0.067**	0.163***	0.154***
	(4.301)	(2.078)	(4.348)	(3.893)
SMB	0.014	-0.007		
	(0.155)	(-0.108)		
HML	0.160**	0.067		
	(2.317)	(0.968)		
Momentum		-0.126***		
		(-3.379)		
Reversal		0.280***		
		(5.212)		
SMB5			0.043	
			(0.622)	
HML5			0.153	
			(1.571)	
RMW5			0.086	
			(1.034)	
CMA5			-0.017	
			(-0.153)	
R_ME				0.009
				(0.096)
R_IA				0.178**
				(2.232)
R_ROE				-0.103
				(-1.265)
R_EG				-0.026
				(-0.297)
R2	0.095	0.305	0.100	0.101
Adj. R2	0.089	0.297	0.091	0.092
Num. obs.	492	492	492	480

* $p < 0.10$, ** $p < 0.05$, and *** $p < 0.01$

Table 5: Size and industry breakdown of the clusters

This table reports the composition of the stocks traded under each clustering method during the test period from 1980.01 to 2020.12. Panel (a) reports the percentage of the stocks in each size decile, where 1 (10) denotes the smallest (largest) stocks. Panel (b) reports the industry sector composition of the stocks. The industry classification is by the North American Industry Classification System (NAICS). The column Market denotes the composition of the entire market. The reported values are time-series averages over the sample period.

(a) Size deciles

	k-means			DBSCAN			Agglomerative		
	L	S	L-S	L	S	L-S	L	S	L-S
10	10.00	12.01	11.01	14.30	16.14	15.22	11.72	13.67	12.69
9	12.44	13.60	13.02	16.80	17.73	17.27	14.60	15.56	15.08
8	12.74	13.90	13.32	15.94	15.79	15.86	14.40	15.32	14.86
7	12.99	13.56	13.28	12.60	13.27	12.94	13.95	14.08	14.02
6	12.23	12.36	12.29	10.91	10.89	10.90	12.19	12.25	12.22
5	10.95	10.72	10.83	8.96	8.66	8.81	10.49	10.05	10.27
4	9.90	9.12	9.51	7.80	7.46	7.63	8.95	8.15	8.55
3	8.52	7.34	7.93	7.28	6.18	6.73	7.17	6.07	6.62
2	6.73	5.23	5.98	4.48	3.33	3.90	5.00	3.88	4.44
1	3.51	2.15	2.83	0.92	0.55	0.73	1.54	0.97	1.25
Average	6.17	6.47	6.32	6.79	6.93	6.86	6.52	6.76	6.64

(b) Industry breakdown

	k-means			DBSCAN			Agglomerative			Market
	L	S	L-S	L	S	L-S	L	S	L-S	
Manufacturing	47.32	47.15	47.23	27.51	26.56	27.04	44.42	44.39	44.41	43.56
Finance, insurance, and real estate	19.48	19.68	19.58	49.84	52.32	51.08	27.61	27.72	27.66	16.06
Services	11.07	11.26	11.17	1.92	2.06	1.99	8.30	8.36	8.33	13.78
Transportation, communications, and utilities	8.24	8.11	8.18	16.24	15.08	15.66	9.08	8.77	8.93	8.79
Retail trade	5.60	5.66	5.63	1.30	1.16	1.23	4.23	4.42	4.32	6.51
Wholesale trade	3.30	3.42	3.36	1.06	0.95	1.00	2.72	2.78	2.75	3.97
Mining	3.07	2.83	2.95	0.79	0.68	0.74	2.08	1.96	2.02	4.37
Construction	0.90	0.87	0.89	0.30	0.42	0.36	0.70	0.72	0.71	1.26
Nonclassifiable establishments	0.68	0.69	0.68	0.71	0.54	0.62	0.61	0.63	0.62	1.23
Agriculture, forestry, and fishing	0.16	0.18	0.17	0.03	0.04	0.03	0.10	0.10	0.10	0.29
Public administration (government)	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02
Number of industries within each cluster	1.31	1.32	1.82	2.73	2.68	3.45	1.07	1.07	1.42	

Table 6: Sub-period analysis

This table reports sub-period performances of the equally-weighted pairs trading portfolios constructed via k-means clustering, DBSCAN, and agglomerative clustering. The sub-periods are defined in the table. The values are annualized where applicable.

	k-means	DBSCAN	Agglomerative	S&P500	Reversal
01/1980 - 12/1999					
Mean return	0.237	0.222	0.258	0.178	0.345
Sharpe ratio	3.554	2.050	3.932	1.088	2.078
Sortino ratio	8.375	3.923	11.660	1.425	3.325
Profit factor	20.019	4.908	20.464	2.301	7.493
Maximum drawdown	-0.040	-0.087	-0.027	-0.314	-0.192
Calmar ratio	5.956	2.563	9.554	0.567	1.799
01/2000 - 12/2006					
Mean return	0.259	0.339	0.297	0.112	0.216
Sharpe ratio	2.352	2.787	2.664	0.683	0.510
Sortino ratio	4.204	9.658	6.052	1.039	0.917
Profit factor	9.252	14.357	13.365	1.669	1.764
Maximum drawdown	-0.080	-0.038	-0.058	-0.299	-0.406
Calmar ratio	3.245	8.864	5.151	0.376	0.532
01/2007 - 12/2009					
Mean return	0.251	0.130	0.235	-0.004	0.165
Sharpe ratio	1.721	0.629	2.113	-0.015	0.671
Sortino ratio	6.235	1.276	5.712	-0.020	1.813
Profit factor	5.511	1.697	7.072	0.988	1.764
Maximum drawdown	-0.083	-0.164	-0.044	-0.551	-0.212
Calmar ratio	3.029	0.790	5.314	-0.007	0.782
01/2010 - 12/2019					
Mean return	0.118	0.229	0.145	0.139	0.037
Sharpe ratio	1.465	1.851	1.783	1.010	0.232
Sortino ratio	3.840	3.275	4.427	1.513	0.486
Profit factor	3.167	4.583	4.372	2.135	1.198
Maximum drawdown	-0.112	-0.123	-0.084	-0.199	-0.260
Calmar ratio	1.053	1.872	1.723	0.698	0.141
01/2020 - 12/2020					
Mean return	0.179	0.513	0.376	0.162	0.511
Sharpe ratio	1.028	2.117	2.191	0.506	1.302
Sortino ratio	2.090	7.578	24.447	0.616	3.421
Profit factor	2.286	5.592	6.938	1.495	3.389
Maximum drawdown	-0.071	-0.060	-0.035	-0.275	-0.159
Calmar ratio	2.512	8.599	10.804	0.590	3.224

Table 7: Parameter sensitivity of the strategies

This table reports the performance of the equally-weighted long-short portfolio constructed from each clustering method with different parameter values. The k-means clustering is tested with different numbers of clusters (K), and DBSCAN and agglomerative clustering are tested with different percentiles (α) for the outlier threshold.

(a) k-means

K	5	10	50	100	500	1000	1500
Mean return	0.175	0.181	0.197	0.200	0.211	0.233	0.274
Sharpe ratio	1.587	1.720	1.946	2.047	2.331	1.762	1.611
Maximum drawdown	-0.173	-0.140	-0.152	-0.141	-0.112	-0.159	-0.236
Number of clusters	4	7	30	51	130	135	81
Number of stocks in clusters	1561 (50.0)	1561 (49.98)	1555 (49.8)	1542 (49.36)	1256 (39.76)	789 (24.23)	390 (10.92)
Number of outliers	1596 (50.0)	1596 (50.02)	1602 (50.2)	1615 (50.64)	1901 (60.24)	2368 (75.77)	2767 (89.08)
Number of stocks traded	525 (16.86)	515 (16.51)	477 (15.29)	459 (14.68)	309 (9.66)	146 (4.3)	51 (1.38)

(b) DBSCAN

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Mean return	0.244	0.190	0.172	0.154	0.149	0.143	0.138	0.131	0.127
Sharpe ratio	1.891	1.669	1.542	1.328	1.195	1.099	1.030	0.936	0.849
Maximum drawdown	-0.164	-0.160	-0.166	-0.179	-0.188	-0.219	-0.231	-0.278	-0.339
Number of clusters	2	2	2	2	2	2	2	2	1
Number of stocks in clusters	377 (12.07)	749 (24.05)	1093 (35.07)	1419 (45.49)	1730 (55.42)	2033 (65.13)	2329 (74.57)	2608 (83.53)	2878 (92.18)
Number of outliers	2780 (87.93)	2408 (75.95)	2064 (64.93)	1738 (54.51)	1427 (44.58)	1124 (34.87)	828 (25.43)	549 (16.47)	279 (7.82)
Number of stocks traded	71 (2.34)	181 (5.94)	306 (9.94)	442 (14.28)	585 (18.8)	734 (23.55)	890 (28.47)	1049 (33.52)	1214 (38.76)

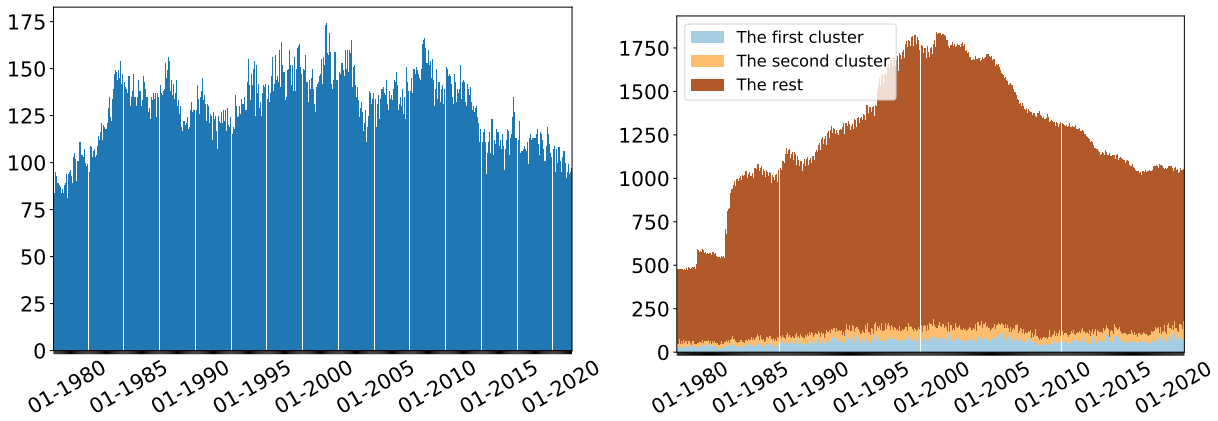
(c) Agglomerative

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Mean return	0.309	0.273	0.249	0.238	0.212	0.201	0.179	0.160	0.142
Sharpe ratio	1.947	2.339	2.593	2.729	2.460	2.238	1.892	1.602	1.222
Maximum drawdown	-0.198	-0.176	-0.098	-0.084	-0.095	-0.128	-0.105	-0.135	-0.202
Number of clusters	80	139	187	223	247	258	252	224	161
Number of stocks in clusters	259 (8.32)	514 (16.44)	768 (24.57)	1030 (32.96)	1304 (41.72)	1591 (50.93)	1899 (60.8)	2234 (71.55)	2605 (83.43)
Number of outliers	2898 (91.68)	2643 (83.56)	2389 (75.43)	2127 (67.04)	1853 (58.28)	1566 (49.07)	1258 (39.2)	923 (28.45)	552 (16.57)
Number of stocks traded	14 (0.46)	47 (1.52)	98 (3.17)	171 (5.49)	265 (8.52)	389 (12.48)	543 (17.39)	736 (23.54)	981 (31.35)

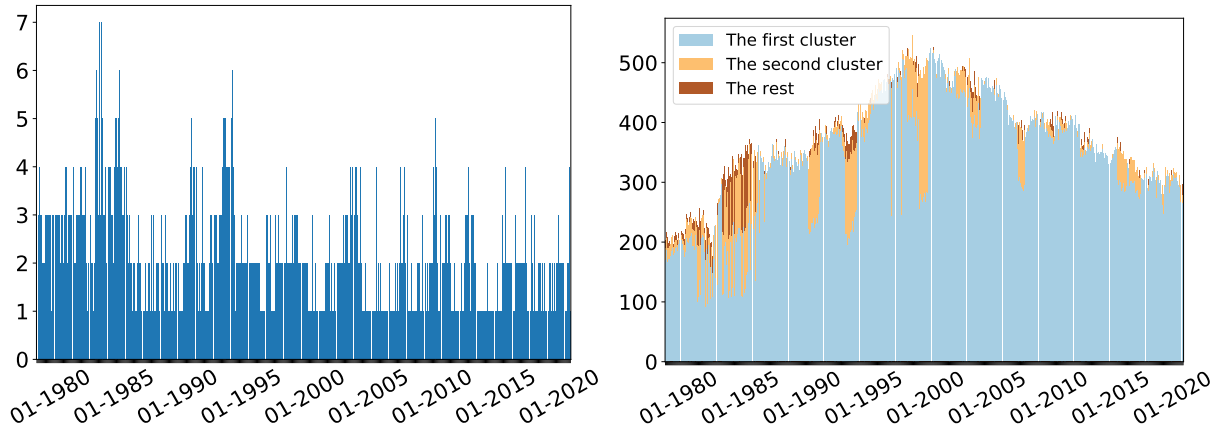
Table 8: Robustness check

This table reports the results from the robustness check in Section 5. 'NYSE10' ('NYSE20') denotes the results after excluding stocks smaller than 10% (20%) NYSE-size quantile, and 'VW' denotes the value-weighted portfolio performance. '10 bps' ('20 bps') denotes the results after considering transaction costs of 10 (20) bps. '0.5 std' ('1.5 std') denotes the results when the trading threshold is set to 0.5 (1.5) times the cross-sectional standard deviation, and '10%' ('5%') denotes the results when the portfolios are constructed from the 10% (5%) highest and lowest return stocks. 'PCA95' ('PCA100') denotes the results when the principal components are chosen so that they can explain 95% (100%) of the variation. The test period is from 1980.01 to 2020.12.

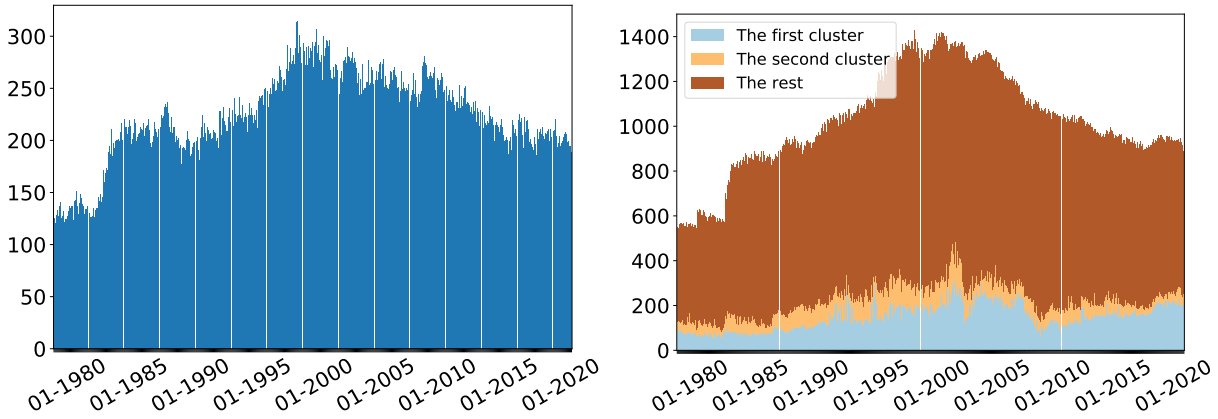
		k-means			DBSCAN			Agglomerative			Benchmarks	
		L	S	L-S	L	S	L-S	L	S	L-S	S&P500	Reversal
NYSE10	MR	0.233	0.067	0.166	0.242	0.038	0.204	0.244	0.060	0.184	0.143	0.239
	SR	1.249	0.420	2.079	1.302	0.251	1.687	1.326	0.392	2.384	0.840	0.983
	MDD	-0.485	-0.513	-0.078	-0.584	-0.528	-0.266	-0.481	-0.482	-0.115	-0.551	-0.503
NYSE20	MR	0.235	0.059	0.176	0.231	0.046	0.184	0.238	0.066	0.171	0.143	0.239
	SR	1.293	0.385	2.100	1.277	0.315	1.504	1.345	0.439	2.239	0.840	0.983
	MDD	-0.474	-0.464	-0.121	-0.542	-0.546	-0.318	-0.425	-0.506	-0.110	-0.551	-0.503
VW	MR	0.179	0.056	0.123	0.197	0.051	0.146	0.201	0.059	0.142	0.126	0.023
	SR	0.936	0.342	1.018	0.980	0.278	0.927	1.002	0.343	1.016	0.835	0.089
	MDD	-0.589	-0.632	-0.224	-0.682	-0.602	-0.556	-0.593	-0.644	-0.248	-0.502	-0.747
10 bps	MR	0.249	0.037	0.168	0.231	-0.015	0.200	0.252	0.012	0.194	0.143	0.239
	SR	1.210	0.224	1.857	1.245	-0.092	1.561	1.329	0.078	2.234	0.840	0.983
	MDD	-0.499	-0.595	-0.138	-0.543	-0.890	-0.170	-0.511	-0.630	-0.105	-0.551	-0.503
20 bps	MR	0.228	0.015	0.124	0.209	-0.038	0.155	0.230	-0.010	0.149	0.143	0.239
	SR	1.104	0.088	1.373	1.125	-0.232	1.211	1.211	-0.065	1.717	0.840	0.983
	MDD	-0.518	-0.622	-0.236	-0.564	-0.939	-0.178	-0.530	-0.789	-0.146	-0.551	-0.503
0.5 std	MR	0.239	0.078	0.161	0.219	0.057	0.162	0.235	0.073	0.163	0.143	0.239
	SR	1.285	0.507	2.460	1.346	0.396	1.812	1.352	0.491	2.777	0.840	0.983
	MDD	-0.482	-0.551	-0.065	-0.500	-0.553	-0.130	-0.466	-0.553	-0.044	-0.551	-0.503
1.5 std	MR	0.298	0.041	0.257	0.279	-0.013	0.291	0.293	-0.004	0.296	0.143	0.239
	SR	1.274	0.223	1.945	1.276	-0.076	1.529	1.309	-0.020	2.217	0.840	0.983
	MDD	-0.456	-0.624	-0.241	-0.674	-0.853	-0.314	-0.518	-0.754	-0.294	-0.551	-0.503
10%	MR	0.242	0.074	0.168	0.247	0.015	0.232	0.217	0.097	0.120	0.143	0.239
	SR	1.270	0.470	2.315	1.357	0.100	1.863	1.256	0.635	2.520	0.840	0.983
	MDD	-0.503	-0.577	-0.075	-0.572	-0.713	-0.201	-0.492	-0.557	-0.053	-0.551	-0.503
5%	MR	0.243	0.077	0.166	0.260	-0.015	0.276	0.215	0.102	0.113	0.143	0.239
	SR	1.216	0.469	2.141	1.275	-0.098	1.752	1.227	0.654	2.382	0.840	0.983
	MDD	-0.530	-0.601	-0.078	-0.661	-0.847	-0.339	-0.498	-0.556	-0.059	-0.551	-0.503
PCA95	MR	0.274	0.059	0.215	0.254	0.008	0.247	0.269	0.044	0.225	0.143	0.239
	SR	1.348	0.359	2.273	1.371	0.047	1.927	1.413	0.279	2.571	0.840	0.983
	MDD	-0.465	-0.588	-0.098	-0.509	-0.821	-0.150	-0.466	-0.566	-0.083	-0.551	-0.503
PCA100	MR	0.263	0.057	0.206	0.250	0.001	0.250	0.277	0.036	0.241	0.143	0.239
	SR	1.255	0.347	2.159	1.333	0.004	1.907	1.462	0.224	2.770	0.840	0.983
	MDD	-0.504	-0.577	-0.198	-0.520	-0.858	-0.176	-0.488	-0.590	-0.059	-0.551	-0.503



(a) k-means



(b) DBSCAN



(c) Agglomerative

Figure 2: Number of clusters and the number of stocks in the clusters

This figure presents the number of clusters and the number of stocks in the clusters formed by k-means clustering, DBSCAN, and agglomerative clustering. The graphs on the left show the number of clusters and the graphs on the right show the number of stocks in the first two biggest clusters every month during the test period from 1980.01 to 2020.12.

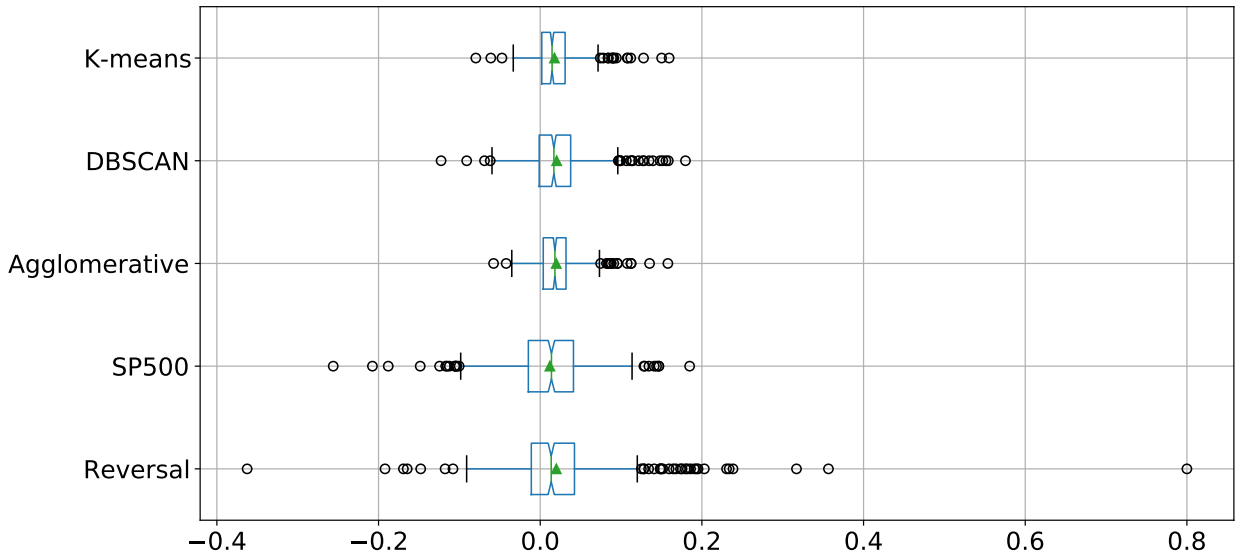


Figure 3: Monthly return distribution

This box plot presents the monthly return distributions of the equally-weighted pairs trading portfolios constructed via k-means clustering, DBSCAN, and agglomerative clustering during the test period from 1980.01 to 2020.12.

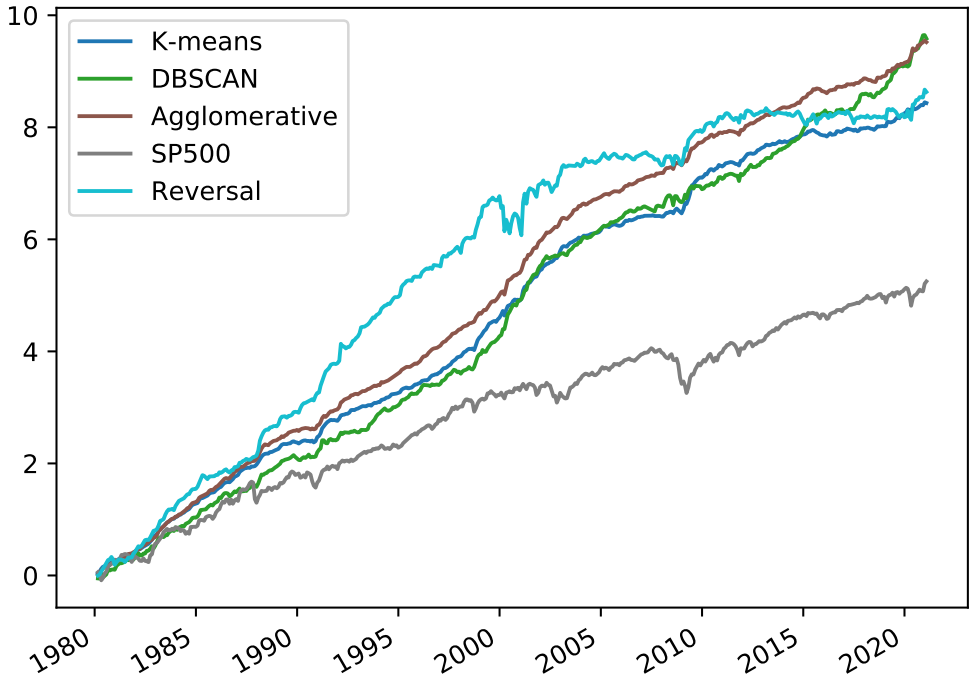


Figure 4: Cumulative returns of pairs trading portfolios

This figure presents the log-scale cumulative returns of the equally-weighted pairs trading portfolios constructed via k-means clustering, DBSCAN, and agglomerative clustering during the test period from 1980.01 to 2020.12.

A Appendix: Firm Characteristics

Acronym	Firm characteristic	Acronym	Firm characteristic
absacc	Absolute accruals	indmom	Industry momentum
acc	Working capital accruals	invest	Capital expenditures and inventory
aeavol	Abnormal earnings announcement volume	IPO	New equity issue
age	# years since first Compustat coverage	lev	Leverage
agr	Asset growth	maxret	Maximum daily return
baspread	Bid-ask spread	MS	Financial statement score
beta	Beta	mve	Size
betasq	Beta squared	mve_ia	Industry-adjusted size
bm	Book-to-market	nincr	Number of earnings increases
bm_ia	Industry-adjusted book to market	operprof	Operating profitability
cash	Cash holdings	pchcapx_ia	Industry adjusted % change in capital expenditures
cashdebt	Cash flow to debt	pchcurrat	% change in current ratio
cashpr	Cash productivity	pchdepr	% change in depreciation
cfp	Cash flow to price ratio	pchgm_pchsale	% change in gross margin - % change in sales
cfp_ia	Industry-adjusted cash flow to price ratio	pchsale_pchrect	% change in sales - % change in A/R
chatoia	Industry-adjusted change in asset turnover	pctacc	Percent accruals
chcsho	Change in shares outstanding	pricedelay	Price delay
chempia	Industry-adjusted change in employees	ps	Financial statements score
chinv	Change in inventory	rd	R&D increase
chmom	Change in 6-month momentum	retvol	Return volatility
chpmia	Industry-adjusted change in profit margin	roaq	Return on assets
chtx	Change in tax expense	roeq	Return on equity
cinvest	Corporate investment	roic	Return on invested capital
convind	Convertible debt indicator	rsup	Revenue surprise
currat	Current ratio	salecash	Sales to cash
depr	Depreciation / PP&E	salerec	Sales to receivables
divi	Dividend initiation	securedind	Secured debt indicator
divo	Dividend omission	sgr	Sales growth
dolvol	Dollar trading volume	sin	Sin stocks
dy	Dividend to price	SP	Sales to price
ear	Earnings announcement return	std_dolvol	Volatility of liquidity (dollar trading volume)
egr	Growth in common shareholder equity	std_turn	Volatility of liquidity (share turnover)
ep	Earnings to price	sue	Unexpected quarterly earnings
gma	Gross profitability	tang	Debt capacity/firm tangibility
herf	Industry sales concentration	tb	Tax income to book income
hire	Employee growth rate	turn	Share turnover
idiovol	Idiosyncratic return volatility	zerotrade	Zero trading days
ill	Illiquidity		

Note: This table lists the 75 firm characteristics used in the study. Readers are referred to Green et al. (2017) and the references therein for the exact definitions.

B Clustering Algorithms

Algorithm 1: k-means clustering

Input: X: All sample points in month t

K: Number of Clusters

$MaxIter$: Maximum number of iterations

Tol: tolerance for convergence

Initialize centroids μ_k's at random points

for i in $MaxIter$ **do**

  for each sample point x in X **do**

    Assign x to its nearest centroid

    Update the centroids μ_k's

  if the l_2 norm of the difference in the cluster centroids of two consecutive iterations is smaller than Tol **then**

    Break

Algorithm 2: Identification of outliers in k-means clustering

Input: X: All sample points in month t

α: Percentile to calculate ϵ, the maximum distance to the centroid for a sample point to remain in its cluster

μ_x: The corresponding centroid of sample point x where $x \in X$

Compute the $l2$ distances of every point to its corresponding centroid

$\epsilon =$ the α percentile of the $l2$ distances

for each sample point x in X **do**

  if $||x - \mu_x||_2 > \epsilon$ **then**

    Mark x as outlier

Algorithm 3: DBSCAN

Input: X: All sample points in month t

$MinPts$: Minimum number of sample points per cluster

ϵ: Maximum distance between two samples for one to be considered the neighbourhood of the other

```
C = 0
for each unvisited sample point x in X do
    Mark x as visited
    NeighborPts = all points within ε from x
    if size of NeighborPts < MinPts then
        Mark x as NOISE
    else
        C = next cluster
        Add x to C
        for each point x' in NeighborPts do
            NeighborPts' = all points within ε from x'
            if size of NeighborPts' ≥ MinPts then
                NeighborPts = NeighborPts ∪ NeighborPts'
```

Algorithm 4: Agglomerative clustering

Input: X: All sample points in month t

ϵ: Maximum distance between two clusters for them to be merged

```
Initialization: Treat all sample points as individual clusters
while minimum distance between clusters < ε do
    Compute the distances between all clusters
    Merge the pair of clusters that has the minimum distance into a single cluster
```