Expert Systems With Applications 195 (2022) 116558

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Asset selection via correlation blockmodel clustering

Wenpin Tang *, Xiao Xu, Xun Yu Zhou

Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, USA

ARTICLE INFO

Keywords:
Asset selection
Cluster analysis

ABSTRACT

We aim to cluster financial assets in order to identify a small set of stocks to approximate the level of diversification of the whole universe of stocks. We develop a data-driven approach to clustering based on a correlation blockmodel, in which assets in the same cluster are highly correlated with each other and, at the same time, have the same correlations with all other assets. We devise an algorithm to detect the clusters, with theoretical analysis and practical guidance. Finally, we conduct an empirical analysis to verify the performance of the algorithm.

1. Introduction

The modern portfolio theory was pioneered by Markowitz (1952, 1959), in which the key insights are diversification and risk–return tradeoff. One drawback of applying Markowitz's mean–variance portfolio selection approach naïvely is to include *all* the available assets for allocation. So in the case of S&P 500, for example, an investor would need to invest in all these 500 stocks in her portfolio. This is simply impossible for small investors or small fund managers.[1] Seasoned investors such as Warren Buffet do not hold many stocks either.[2] Even gigantic funds such as Vanguard and BlackRock do not include, even though they could, all the stocks in their portfolios. Practically, managing too many stocks is costly and prone to mismanagement. According to a Morningstar article, "when you lose your focus and move outside your circle of competence, you lose your competitive advantage".[3] Technically, including too many stocks increases both the odds of overfitting and the difficulty in computing efficient allocation strategies (e.g., DeMiguel, Garlappi, & Uppal, 2009). One way to address this issue is to add a regularization term or a cardinality constraint in the Markowitz mean–variance optimization model (Brodie et al., 2009; DeMiguel, Garlappi, Nogales, & Uppal, 2009; Faaland, 1974; Gao & Li, 2013; Ho et al., 2015). This approach imposes sparsity on the number of assets in the portfolio; however, the regularization itself does not take diversification into account, and thus the set of stocks selected may contain concentration risk.

Since the main reason to include all the stocks is to diversify, we have the following natural question: How can we select a *much* smaller subset of the whole universe of stocks that achieves a *sufficient* level of diversification?[4] Reilly and Brown (2012) states that "about 90% of the maximum benefit of diversification was derived from portfolios of 12 to 18 stocks". Markowitz (1952) suggests a simple rule of thumb for selecting stocks that one should try to "diversify across industries because firms in different industries, especially industries with different economic characteristics, have lower covariances than firms within an industry". In practice, stock selection is often based on factors such as sector rotation and macroeconomic indicators and is subjective to each investor. However, this approach relies on the taxonomy of sectors and macroeconomic analysis published by certain organizations, which may be subject to different interpretations and may contain biases. If, as noted above, the primary goal of stock selection is to achieve sufficient diversification to which asset correlations are the key, approaches focusing directly on asset correlations are more appropriate and more innately fitting for the subsequent asset allocation. A promising such approach is *clustering* based on correlation networks. Specifically, one first groups or clusters all the assets in a correlation network and then

* Corresponding author.
E-mail addresses: wt2319@columbia.edu (W. Tang), xx2167@columbia.edu (X. Xu), xz2574@columbia.edu (X.Y. Zhou).

[1] Some investment experts suggest that 30 stocks be the maximum number of stocks in a retail investor's stock portfolio ("How Many Stocks Should Be in a Portfolio?", Zacks, 2019, Accessed January 5, 2021. https://finance.zacks.com/many-stocks-should-portfolio-4782.html).

[2] Between Berkshire Hathaway and New England Asset Management, Buffet holds 49 stocks in total, with about 92% of the portfolio concentrating in 15 stocks, and 78% in just five stocks (based on holdings as of September 30, 2020, reported in Berkshire Hathaway and New England Asset Management's 13F filings on November 16, 2020).

[3] "How Many Stocks Diversify Unsystematic Risk?", Morningstar, Accessed January 5th, 2021. http://news.morningstar.com/classroom2/course.asp?docId=145385&page=4.

[4] If this question can be satisfactorily answered, we can then apply Markowitz's mean–variance model to this small set of stocks to get an efficient portfolio. In other words, we can decompose the Markowitz model into two stages: asset *selection* and asset *allocation*.

https://doi.org/10.1016/j.eswa.2022.116558
Received 28 August 2021; Received in revised form 29 December 2021; Accepted 16 January 2022
Available online 5 February 2022

selects one or a few "representative" assets in each group, resulting in a subset of a much smaller number of assets.

Since the seminal work of Mantegna (1999), correlation networks have been widely used as a tool to study the correlation structure of financial assets. In a correlation network, financial assets are modeled as nodes, connected by edges representing the correlations between their returns. Clustering analysis is conducted on correlation networks, and clusters are often compared with traditional industry classifications (Musmeci et al., 2015; Rosén, 2006). One line of the clustering research is simply to understand the market structure without involving portfolio selection; see Das (2003), Marathe and Shawky (1999) with k-means algorithm, Gavrilov et al. (2000) with hierarchical clustering, and Mantegna (1999), Tumminello et al. (2005) with network filtering. Another line of research is to utilize the revealed market structure to construct portfolios. For instance, Ren (2005) creates clusters based on a simple threshold rule and constructs an optimal portfolio of sub-portfolios, each of which is an equally-weighted portfolio of all stocks in the same cluster. Based on just the structure of the correlation networks, Pozzi et al. (2013) build portfolios consisting of stocks in the center and on the periphery of the networks. For various studies applying the same idea of "clustering analysis and portfolio construction using representative sub-portfolios", see, e.g., De Prado (2016), Korzeniewski (2018), León et al. (2017), Marvin (2015), Nanda et al. (2010), Raffinot (2017), Zhan et al. (2015). Most recently, Puerto et al. (2020) propose a unified framework that pursues high within-cluster correlations while at the same time optimizing the allocations of the representative stocks.

Despite their popularity in the machine learning community, the k-means algorithm (Lloyd, 1982) and the similar k-medoids algorithm (or Partitioning Around Medoids, "PAM", Kaufman & Rousseeuw, 1990) have found only limited applications to finance. Marvin (2015) applies k-medoids to cluster financial assets, albeit based on financial ratios of companies instead of return time series. Musmeci et al. (2015) compare k-medoids with other clustering methods along with the industry classification. He et al. (2007) and Nakagawa et al. (2019) apply k-means and k-medoids to financial time series data for a different purpose: they group series of returns in order to predict future returns.

Most of the above clustering methods applied to portfolio selection are heuristics, hence often difficult to interpret. In this paper, we propose a new, interpretable, data-driven approach to correlation network clustering and provide a systematic solution for selecting well-diversified stocks. Our clustering is based on the following two criteria:

Criterion 1. *Financial assets in the same group have high correlations.*

Criterion 2. *Financial assets in the same group have similar correlations with all other assets.*

Criterion 1 is self-evident. Assets with high correlations may perform well simultaneously at one time and plummet simultaneously at another time; so we cluster them into the same group. The next important question is how to select the "representative" assets in each group of a clustering. Practitioners often choose the best assets in each group according to their own performance metrics. However, this does not guarantee that the assets selected in each group are "optimal" especially in terms of their relationship with other assets outside of the group. Motivated by the mean–variance portfolio theory, we propose to cluster in such a way that any two assets in the same group have similar correlations to *all* others in the stock universe, which underlines Criterion 2. So any two assets in the same group are interchangeable in terms of their correlations with other assets. Consequently, one needs to only choose some idiosyncratic characteristics, such as volatility or Sharpe ratio, in selecting which asset to be included in the portfolio. This makes the choice of representative assets from each cluster simple and transparent. The idea, though very natural, seems missing in the literature.

The purpose of this paper is to develop a new financial clustering approach, taking *both* criteria into account. We propose a *correlation blockmodel* to capture Criterion 2. This formulation is inspired by the problem of community detection in stochastic blockmodel (Abbe, 2017) and block covariance model (Bunea et al., 2016, 2020). In the model, the return of any asset in the same group is expressed as the sum of a common latent factor and an uncorrelated random noise. As such, any two assets in the same group have the same correlations to all others. Criterion 1 is then used to calibrate a threshold hyperparameter that controls how variables are grouped together. We devise an algorithm – called ACC (Asset Clustering through Correlation) – to recover the clusters of the blockmodel in polynomial time.

The main contributions of this work are as follows. First, to our best knowledge, this paper is the first to implement both criteria (especially Criterion 2) in financial asset clustering to capture the notion of diversification and the first to utilize the correlation blockmodel to formalize the implementation. This provides interpretability of our clustering approach from the portfolio theory point of view. Second, we lay a rigorous *theoretical* foundation for the clustering algorithm from both algorithmic and statistical perspectives. In particular, we provide a statistical guarantee for the algorithm which can account for the possible heavy-tailed data intrinsic to financial time series. This estimate requires a delicate analysis owing to the heavy tails and is new and interesting in its own right. Moreover, we propose a hyperparameter tuning procedure in the clustering algorithm, taking both criteria into account. The information limit of the blockmodel narrows down the search for the hyperparameter, while Criterion 1 is used to cross-validate. Finally, we conduct an extensive empirical study on the S&P 500 stocks by selecting 15 to 25 stocks at a time via clustering and constructing portfolios using the selected stocks. For comparison, we select stocks from clusters created by the popular k-medoids clustering algorithm and clusters based on S&P's sector and industry classification. We also consider the set of all S&P 500 sector ETFs, each of which represents a different sector in the S&P 500 Index. For all these groups of stocks, we employ and compare three asset allocation strategies: risk parity, minimum-variance, and Markowitz's mean–variance optimal allocation. The results show that the portfolios constructed using our ACC algorithm outperform the benchmark – the S&P 500 ETF – significantly. The portfolios based on ACC clusters also perform favorably compared to all other portfolios, especially when portfolios are readjusted infrequently.

In Bunea et al. (2016) (which is an unpublished, earlier version of Bunea et al., 2020), an algorithm is presented to recover the clusters under the same correlation blockmodel. Our results and algorithm differ significantly from Bunea et al. (2016) in the following aspects. First, while Bunea et al. (2016) briefly demonstrate their model by applying the clustering algorithm to stock data in their numerical experiments, we are motivated by the modern portfolio theory to construct and justify the model and offer a theoretical interpretation of the model related to diversification based on the two criteria. Second, we employ a different tuning procedure in order to incorporate Criterion 1. Third, the underlying distribution is restricted to Gaussian in Bunea et al. (2016), while we consider a more general range of distributions that encompass the heavy-tails prevalent in financial data. Lastly, we conduct portfolio construction and extensive backtesting, which is not the primary focus of Bunea et al. (2016).

The remainder of the paper is organized as follows. In Section 2, we present the correlation blockmodel and the ACC algorithm and state the main theoretical results. Section 3 provides an empirical analysis. We conclude with a few remarks in Section 4. All the proofs are contained in Appendix.

2. Correlation blockmodel and clustering algorithm

We first collect some notations that will be used throughout this paper. All vectors are column vectors unless stated otherwise.

- We use bold case letters, e.g., $\boldsymbol{X}$, to denote matrices.
- For a vector x, $|x|$ is the Euclidean norm of x.
- For a vector x (resp. a matrix $\boldsymbol{X}$), $x^\top$ (resp. $\boldsymbol{X}^\top$) is the transpose of x (resp. $\boldsymbol{X}$).
- For a set A, $|A|$ is the number of elements in A.
- For a random variable X, $\mathbb{E}(X)$ is the expectation of X, and $\mathrm{Var}(X)$ the variance of X.
- For two random variables X and Y, $\mathrm{Cov}(X,Y) := \mathbb{E}[(X-\mathbb{E}(X))(Y-\mathbb{E}(Y))]$ is the covariance between X and Y, and
$$\mathrm{Corr}(X,Y) := \frac{\mathbb{E}[(X-\mathbb{E}(X))(Y-\mathbb{E}(Y))]}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}},$$
is the Pearson correlation coefficient between X and Y.
- a and b have the same order of magnitude, denoted by $a \asymp b$, if and only if $c \le a/b \le C$ for some $c, C > 0$, as $a, b \to \infty$.

2.1. Model setup

Assume that there are d financial assets, indexed by $[d] := \{1, \dots, d\}$. For $i \in [d]$, let X_i be the return of asset i, and $X_i^* := (X_i - \mathbb{E}(X_i))/\sqrt{\mathrm{Var}(X_i)}$ be the standardized return. Throughout this paper we assume shorting is not allowed; hence the return of an asset refers to that of a *long* position. As mentioned in the introduction, one of our goals is to cluster the returns $X = (X_1, \dots, X_d)^\top$ in such a way that X_i and X_j belong to the same group if and only if they have the same correlations with *all* other returns, i.e., $\mathrm{Corr}(X_i, X_l) = \mathrm{Corr}(X_j, X_l)$ for $l \neq i, j$. This amounts to finding a partition $G = \{G_1, \dots, G_K\}$ of $[d]$, or a map of membership assignment $z : [d] \to [K]$ so that

$$X_i \text{ belongs to group } k \iff i \in G_k \iff z(i) = k.$$

The sets $G_1, \dots, G_K$ are called the *blocks*, or the groups of the partition G, which define an equivalence relation $i \overset{G}{\sim} j$ if and only if $i, j \in G_k$ for some $k \in [K]$, or simply $z(i) = z(j)$. Similarly, $i \overset{G}{\nsim} j$ if and only if there is no block G_k that contains both i and j, or $z(i) \neq z(j)$.

We now introduce the correlation blockmodel, in which X_i's in the same group can be decomposed as the sum of a common latent factor and an uncorrelated random fluctuation. Precisely, the standardized returns are represented as

$$X_i^* = F_{z(i)} + U_i, \quad i \in [d], \tag{1}$$

where

- $F = (F_1, \dots, F_K)$ are the latent factors with $\mathbb{E}(F_k) = 0$ for each $k \in [K]$;
- $U = (U_1, \dots, U_d)$ are idiosyncratic fluctuations with $\mathbb{E}(U_i) = 0$ for each $i \in [d]$, $\mathrm{Cov}(U_i, U_j) = 0$ for $i \neq j$, and $\mathrm{Cov}(F_k, U_i) = 0$ for each k, i.

It is easy to see that for $i \overset{G}{\sim} j$ and $l \neq i, j$, $\mathrm{Corr}(X_i, X_l) = \mathbb{E}(F_{z(i)}F_{z(l)}) = \mathbb{E}(F_{z(j)}F_{z(l)}) = \mathrm{Corr}(X_j, X_l)$. Moreover, let $\sigma_k^2 := \mathrm{Var}(F_k)$ be the variance of the latent factor underlying group $k \in [K]$. By definition (1), $\mathrm{Var}(U_i) = 1 - \sigma_k^2$ if $i \in G_k$. This implies that the signal-to-noise ratios are the same for all standardized returns belonging to the same group.

Given such a correlation blockmodel, the question is to infer the block structure – the partition G from the correlation matrix $\boldsymbol{\rho} := \mathbb{E}(X^* X^{*\top})$, where $X^* = (X_1^*, \dots, X_d^*)^\top$ are standardized returns. Denote $\boldsymbol{\Sigma}_F := \mathbb{E}(FF^\top)$ and $\boldsymbol{\Sigma}_U := \mathbb{E}(UU^\top)$ as the covariance matrices of F and U respectively. Here $\boldsymbol{\Sigma}_U$ is diagonal, since $\mathrm{Cov}(U_i, U_j) = 0$ for $i \neq j$. Let $\boldsymbol{Z} := (1_{\{z(i)=k\}})_{(i,k)\in[d]\times[K]}$ be the membership matrix. The correlation matrix $\boldsymbol{\rho}$ is then expressed as

$$\boldsymbol{\rho} = \boldsymbol{Z}\boldsymbol{\Sigma}_F\boldsymbol{Z}^\top + \boldsymbol{\Sigma}_U. \tag{2}$$

Note that the partition G which satisfies (1) or (2) may not be unique. This could be due to overly granular partitions splitting the set of returns with the same latent factor further into smaller groups, or due to different latent factors that share the same covariances with all other latent factors. A natural way to address this problem is to look for the *coarsest* partition, which has the least number of clusters. Since the partition order is a partial order, the coarsest partition is, in general, still not necessarily unique. Nevertheless, the following result ensures a unique coarsest partition $G^\star$ for the correlation blockmodel (1).

Theorem 1. *Let $\boldsymbol{\rho}$ be a correlation matrix. Then there is a unique coarsest partition $G^\star$ such that $\boldsymbol{\rho} = \boldsymbol{Z}\boldsymbol{\Pi}\boldsymbol{Z}^\top + \boldsymbol{\Gamma}$ for some membership matrix $\boldsymbol{Z}$ associated with $G^\star$, some matrix $\boldsymbol{\Pi}$, and some diagonal matrix $\boldsymbol{\Gamma}$. Moreover, the partition $G^\star$ is defined by the equivalence relation*

$$i \overset{G^\star}{\sim} j \quad \textit{if and only if} \quad \max_{l\neq i,j} |\rho_{il} - \rho_{jl}| = 0. \tag{3}$$

The proof is deferred to Appendix A.1. Theorem 1 shows that the coarsest partition $G^\star$ is well-defined; so the clusters of the correlation blockmodel (2) are identifiable. We note that the coarsest partition could potentially group two clusters controlled by different factors as one, if the factors have the same covariances with all other factors. This is not a problem for our purposes, as this partition would still satisfy Criterion 2. In the remainder of this work, we aim to recover or to estimate the partition $G^\star$ from historical data.

2.2. The PARTITION procedure

According to Theorem 1, two financial asset returns X_i and X_j belong to different clusters in G^* if and only if $\max_{l\neq i,j} |\rho_{il} - \rho_{jl}| > 0$. This observation motivates the definition of a dissimilarity measure between assets i and j – correlation difference (CORD, Bunea et al., 2016):

$$\mathrm{CORD}(i,j) := \max_{l\neq i,j} |\rho_{il} - \rho_{jl}|, \quad i, j \in [d]. \tag{4}$$

This measure quantifies the dissimilarity between two assets in terms of their respective correlations with *all* other assets. Consider a set of financial asset returns that includes X_i and some other returns $Y_1, Y_2, \dots$. If we position X_i at the top, the corresponding row in the covariance matrix of these returns is:

$$\Sigma_i = (\mathrm{Var}(X_i), \mathrm{Cov}(X_i, Y_1), \mathrm{Cov}(X_i, Y_2), \dots).$$

If we have $\mathrm{CORD}(i,j) = 0$, then $\mathrm{Cov}(X_j, Y_k) = c\,\mathrm{Cov}(X_i, Y_k)$ for all $k = 1, 2, \dots$, where $c = \sqrt{\mathrm{Var}(X_j)/\mathrm{Var}(X_i)}$ is a constant. So if we replace X_i with X_j in the set of asset returns, the first row (and column) of the covariance matrix will only be rescaled by a constant factor c, except for the variance term, which will be scaled by c^2. Then in a minimum-variance portfolio with no short selling, which optimizes the weights to minimize the portfolio variance based on the covariance matrix, the assets i and j are interchangeable up to a constant factor.[5] This interchangeability also leads to the following result, which will be proved in Appendix A.2.

Theorem 2. *Under the correlation blockmodel with a coarsest partition $G^* = \{G_1, G_2, \dots, G_K\}$, construct a minimum variance portfolio by choosing one asset from each cluster:*

$$P_J := \{J(1), J(2), \dots, J(K)\},$$

[5] Recall that a no-short-selling minimum variance portfolio of a set of assets with covariance matrix $\boldsymbol{\Sigma}$ is one with asset weights w that solves the optimization problem:

$$\begin{aligned} \min \quad & w^\top \boldsymbol{\Sigma} w \\ \text{s.t.} \quad & w^\top 1 = 1 \\ & w \ge 0. \end{aligned}$$

where $J(k) \in G_k$, for $k = 1, \dots, K$. Among all such portfolios P_J, the portfolio with the lowest variance is the one consists of the asset with the lowest variance in each cluster: $\operatorname{argmin}_J \operatorname{Var}(P_J) = \{J^*(1), J^*(2), \dots, J^*(K)\}$ *where*

$$J^*(k) = \operatorname*{argmin}_{j \in G_k} \operatorname{Var}(X_j), \quad \forall k = 1, \dots, K. \tag{5}$$

This theorem gives guidance on choosing an asset from each cluster to attain the minimum variance among all possible minimum variance portfolios. Arguably, this selection approach aligns with Markowitz's original notion that diversification can be measured by variance minimization.

Next, we discuss how to derive clustering from data. Assume that the financial asset returns are observed over n periods. For $r \in [n]$, let $X^r = (X_1^r, \dots, X_d^r)^\top$ be the asset returns in period r, and $X^{*r} = (X_1^{*r}, \dots, X_d^{*r})^\top$ be the corresponding standardized returns. Denote by $\boldsymbol{X}^*$ the $n \times d$ matrix whose row r is X^{*r}. Also, assume that $X^1, \dots, X^n$ are independent and identically distributed (i.i.d.) so that $X^{*1}, \dots, X^{*n}$ are i.i.d. copies of X^* defined by (1). The goal is to estimate the cluster partition $G^\star$ from the sample correlation matrix $\widehat{\boldsymbol{\rho}}$ given by

$$\widehat{\boldsymbol{\rho}} := \frac{1}{n-1}(\boldsymbol{X}^*)^\top \boldsymbol{X}^* = \frac{1}{n-1}\sum_{r=1}^{n} X^{*r}(X^{*r})^\top. \tag{6}$$

Define the sample correlation difference $\widehat{\mathrm{CORD}}$ by

$$\widehat{\mathrm{CORD}}(i,j) := \max_{l \neq i,j} |\widehat{\rho}_{il} - \widehat{\rho}_{jl}|, \quad i, j \in [d]. \tag{7}$$

Given the sample correlation differences, the clusters can be recovered through an iterative procedure that takes a dissimilarity matrix $\boldsymbol{D}$ and a threshold parameter ε as inputs. This procedure, which we call PARTITION, is a generalization of the CORD algorithm (Bunea et al., 2016) and is described as Procedure 1. The main idea of the PARTITION procedure is that two assets i and j should belong to the same cluster if their dissimilarity, denoted by $\boldsymbol{D}(i,j)$, is small, e.g., below a threshold $\varepsilon > 0$. In each iteration, the procedure identifies a new cluster by finding the most similar pair of assets, i.e., with the smallest $\boldsymbol{D}(i,j)$. If $\boldsymbol{D}(i,j)$ between these two assets is lower than the predetermined threshold ε, then the two assets act as the core of the cluster, and all other assets that are similar to *either* of the core assets are included in the cluster. Otherwise, any one of the two assets is singled out as its own cluster. Let us note that the PARTITION procedure does not require as input the number of clusters K, which is determined via the threshold ε. In our method, we will use the sample correlation difference $\widehat{\mathrm{CORD}}$ between assets as the input to recover the clusters under the correlation blockmodel. We will explain the method to determine the appropriate value for the threshold ε in the upcoming sections.

Note that the PARTITION procedure bears some similarity to the classical single-linkage clustering procedure (Anderberg, 1973; Mantegna, 1999) in that they are both conglomerative, namely clusters are built by grouping elements together, as opposed to being divisive where clusters are created through separations. However, the two are different in more fundamental ways. Single-linkage is a *hierarchical* algorithm. At each iteration, it merges two existing (possibly singleton) clusters by the shortest linkage between nodes in different clusters. If done all the way to the very end, all nodes will be merged into a single big cluster; so the algorithm creates a hierarchical structure represented by a tree-like dendrogram. The desired number of clusters is obtained by stopping the algorithm early, or by observing the dendrogram. In contrast, the PARTITION procedure does not create such a hierarchy, because it does not merge clusters together. After an iteration, all nodes are either unclustered or already clustered, and the next iteration creates a new cluster out of the unclustered elements. PARTITION also cannot explicitly take in the number of clusters as an input parameter: the number of clusters is endogenously determined by the threshold ε.

We now study the statistical property of the PARTITION procedure when applied to the sample correlation difference. To do this, we need the following assumption on the distribution of the asset returns.

Procedure 1 Partition.

procedure PARTITION($\boldsymbol{D}$, ε) ▷ $\boldsymbol{D}$ is a given dissimilarity matrix; $\varepsilon > 0$
  Initialization: $S \longleftarrow [d]$, $l \longleftarrow 0$.
  while $S \neq \emptyset$ **do**
    $l \longleftarrow l + 1$
    if $|S| = 1$ **then**
      $\widehat{G}_l \longleftarrow S$
    if $|S| > 1$ **then**
      $(i_l, j_l) \longleftarrow \operatorname{argmin}_{i,j \in S, i \neq j} \boldsymbol{D}(i,j)$
      if $\boldsymbol{D}(i_l, j_l) > \varepsilon$ **then**
        $\widehat{G}_l \longleftarrow \{i_l\}$
      else
        $\widehat{G}_l \longleftarrow \{k \in S : \min\left(\boldsymbol{D}(i_l,k), \boldsymbol{D}(j_l,k)\right) \leq \varepsilon\}$
    $S \longleftarrow S \setminus \widehat{G}_l$
  return $\widehat{G} = \{\widehat{G}_1, \widehat{G}_2, \dots\}$

Assumption 1. The correlation matrix $\boldsymbol{\rho}$ is non-singular, and the vector $\boldsymbol{\rho}^{-1/2}X^*$ is α-sub-exponential, $\alpha \in (0, 2]$; that is, there exists $L > 0$ such that

$$\|\boldsymbol{\rho}^{-1/2}X^*\|_{\psi_\alpha} \leq L,$$

where $\|Z\|_{\psi_\alpha} := \sup_{\|\omega\|_2 = 1} \inf\{s > 0 : \mathbb{E}(e^{(|Z^\top \omega|/s)^\alpha}) \leq 2\}$ is the α-Orlicz norm of $Z \in \mathbb{R}^d$.

The non-singularity amounts to the non-existence of redundant securities, i.e., there does not exist any asset whose return is a linear combination of those of the other assets in the universe. The α-sub-exponential distribution was introduced in Krasnoselsky and Rutitsky (1961) to characterize heavy-tailed random variables. The special cases $\alpha = 2$ and $\alpha = 1$ correspond to the sub-Gaussian and sub-exponential variables, respectively. The lower α is, the more heavy-tailedness is allowed. Assumption 1 is motivated by the stylized fact that financial assets often have heavy-tailed returns; see, e.g., Cont (2001).

The following result provides the statistical guarantee for the PARTITION procedure, along with guidance for the choice of the threshold ε.

Theorem 3. *Under Assumption 1, there exist numerical constants $c_1, c_2 > 0$ independent of n and d, such that if* $\min_{i \overset{G^\star}{\nsim} j} \mathrm{CORD}(i,j) > \varepsilon$ *and*

$$\varepsilon \geq 2L^2\left(c_1\sqrt{\frac{\log d}{n}} + c_2\frac{(\log d)^{2/\alpha}}{n}\right), \tag{8}$$

then the PARTITION procedure with inputs $\widehat{\mathrm{CORD}}$ and ε outputs $\widehat{G} = G^\star$ with probability $1 - 4/d$.

The proof of Theorem 3 is deferred to Appendix A.3. Theorem 3 implies that under a cluster separation condition and when the number of variables d is large, the PARTITION procedure recovers the clusters with high probability if the threshold ε is roughly of order $\max(\sqrt{\log d\,/n}, (\log d)^{2/\alpha}/n)$. When $d = 500$ for instance, this probability is 99.2%.

Notice that $\sqrt{\log d\,/n}$ dominates $(\log d)^{2/\alpha}/n$ if $n > (\log d)^{\frac{4}{\alpha}-1}$. In practice, the number of observations n is the same order as the number of assets d. So as $n \asymp d \to \infty$, the right hand side of (8) is dominated by $\sqrt{\log d\,/n}$. However, the comparison of these two terms is sensitive to the value of α when $n \asymp d$ and both are finite. For instance, consider a universe of $d = 500$ financial assets. Table 1 displays the values of n above which $\sqrt{\log d\,/n} > (\log d)^{2/\alpha}/n$ for different α. Therefore, it is important to accurately estimate α in order to determine which of the two terms, $\sqrt{\log d\,/n}$ and $(\log d)^{2/\alpha}/n$, dominates.

Table 1
Values of n above which $\sqrt{\log d\,/n} > (\log d)^{2/\alpha}/n$ for $d = 500$.

α	0.25	0.5	0.75	1.0	1.25	1.5	1.75	2.0
n	7.97×10^{11}	3.58×10^{5}	2.74×10^{3}	240.02	55.66	21.01	10.47	6.21

2.3. Tuning the threshold ϵ

The effectiveness of the PARTITION procedure depends on the threshold ϵ, which in turn determines the number of clusters K. For instance, a low threshold, e.g., $\epsilon = 0$, leads to many singleton clusters due to noise in the observations. On the other hand, a high threshold, e.g., $\epsilon = 2$, results in a single cluster because $\widehat{\text{CORD}}$, which is the maximum among differences between sample correlations, has a maximum value of 2. Theorem 3 provides a part of the guidance for choosing a suitable ϵ. Here, we propose a data-driven approach to tune the hyperparameter ϵ based on the following three rules of thumb.

First of all, according to Theorem 3, the PARTITION procedure recovers the partition $G^\star$ if ϵ is of order $L^2 \max(\sqrt{\log d\,/n}, (\log d)^{2/\alpha}/n)$. This criterion gives a reasonable range for the choice of ϵ.

Rule 1. *The search range for the threshold ϵ is determined as follows:*

1. *If $n > (\log d)^{\frac{4}{\alpha}-1}$, then the range is set to be $[a, b] \times L^2 \sqrt{\frac{\log d}{n}}$;*
2. *If $n \le (\log d)^{\frac{4}{\alpha}-1}$, then the range is set to be $[a, b] \times L^2 \frac{(\log d)^{2/\alpha}}{n}$.*

Here a, b are user-defined parameters.

With the sample correlation difference as input, the PARTITION procedure captures Criterion 2. However, it does not take into account Criterion 1—financial assets in the same cluster are highly correlated. To incorporate this criterion, we propose to calibrate the value of ϵ by considering the intra-cluster correlations. To be more precise, let $\widehat{G}_\epsilon$ be the output given by the PARTITION procedure with a threshold ϵ. Define the intra-cluster correlation $\hat{\rho}_\epsilon^{ave}$ of $\widehat{G}_\epsilon$ by

$$\hat{\rho}_\epsilon^{ave} := \frac{\sum_{i<j} 1\left(i \overset{\widehat{G}_\epsilon}{\sim} j\right)\hat{\rho}_{ij}}{\sum_{i<j} 1\left(i \overset{\widehat{G}_\epsilon}{\sim} j\right)}, \tag{9}$$

where $\hat{\rho}_{ij}$ is the sample correlation between asset returns i and j given by (6). The goal is to select the threshold ϵ that gives the maximal intra-cluster correlation.

Rule 2. *Let $\mathcal{T}$ be the range specified in Rule 1. We choose*

$$\epsilon_{\mathcal{T}} := \operatorname*{argmax}_{\epsilon \in \mathcal{T}} \hat{\rho}_\epsilon^{ave}.$$

One disadvantage of applying Rule 2 naïvely is that it is biased towards granular partitions with very few but high intra-cluster correlations. Specifically, consider an extreme case where each asset forms its own cluster except for two assets i and j with correlation $\rho_{ij} = 1$, which form one cluster $\{i, j\}$. By definition (9), the average intra-cluster correlation under this partition is $\hat{\rho}^{ave} = \rho_{ij} = 1$. However, such a partition that only groups two assets i and j together is not very informative, despite indeed being optimal under Rule 2. To regularize Rule 2, we propose to put constraints on the number of clusters. That is, we set a range for the number of clusters. Thresholds resulting in too many or too few clusters are discarded, and then the one with the highest intra-cluster correlation is selected. This results in the final rule:

Rule 3. *Let $\mathcal{T}$ be the range specified in Rule 1, and $\mathcal{U}$ be a user-defined range for the number of clusters. We choose*

$$\epsilon_{\mathcal{T}\mathcal{U}} := \operatorname*{argmax}_{\epsilon \in \mathcal{T}, |\widehat{G}_\epsilon| \in \mathcal{U}} \hat{\rho}_\epsilon^{ave}.$$

In solving the above maximization problem, the grid search is performed on the search range, split into N grids, where N is a user-defined parameter for the search.

2.4. Estimation of heavy-tailedness

By Rule 1, the heavy-tailedness α together with some constant L that depends on α determines the range in which we search for the threshold ϵ. Thus, we need to estimate the parameter α and constant L, which encode the heavy-tailed nature of the returns X^*.

First, Vladimirova et al. (2020) prove that Assumption 1 is equivalent to:

$$\exists L > 0 \text{ such that } \mathbb{P}\left(\left|\left(\boldsymbol{\rho}^{-1/2}X^*\right)^\top w\right| > t\right) \le 2\exp\left(-(t/L)^\alpha\right) \quad \text{for } \forall t \ge 0 \text{ and } \forall w \in \mathbb{R}^d, \|w\|_2 = 1,$$

where the L values are the same as in Assumption 1.

In order to facilitate the estimation of the tail parameters L and α, we further restrict w to singleton vectors,[6] i.e., vectors with only one element being 1 while the rest being 0. This transforms the assumption to, for some $\alpha \in (0, 2]$:

$$\exists L > 0 \text{ such that } \mathbb{P}\left(\left|\left(\boldsymbol{\rho}^{-1/2}X^*\right)_r\right| > t\right) \le 2\exp\left(-(t/L)^\alpha\right) \quad \text{for } \forall t \ge 0 \text{ and } \forall r \in [d], \tag{10}$$

where $(\boldsymbol{\rho}^{-1/2}X^*)_r$ is the rth coordinate of the random vector $\boldsymbol{\rho}^{-1/2}X^*$. Recall that the smaller α is, the more heavy-tailedness is allowed in the distribution. So if X^* satisfies (10) for some $\alpha > 0$, it also does for all $\alpha' \in (0, \alpha)$. Similarly, for any fixed α, if the inequality in (10) holds for some L, it also holds for any $L' > L$. Thus, we aim to find the largest α such that (10) holds for all $r \in [d]$, and the smallest L for such α. Notice that (10) is trivially satisfied for small t, since when $t < L(\log 2)^{\frac{1}{\alpha}}$, the right hand side is greater than 1. This means that Assumption 1 only controls the tail distribution of $|(\boldsymbol{\rho}^{-1/2}X^*)_r|$. Given some α and L, consider the tail distribution characterized by the following survival function:

$$\mathbb{P}(Y > t) = 2\exp\left(-(t/L)^\alpha\right) \quad \text{for } t \ge L(\log 2)^{\frac{1}{\alpha}}. \tag{11}$$

Loosely speaking, this distribution concerns the *boundary* of the condition (10). If for each coordinate r in the random vector, we can find suitable α_r and L_r such that the distribution (11) fits the tail observations of the vector $Y_r := |(\boldsymbol{\rho}^{-1/2}X^*)_r|$, then $\alpha^* := \min_{r\in[d]} \alpha_r$ and $L^* := \max_{r\in[d]} L_r$ estimate the largest α and the smallest L that satisfy the inequality (10) for all $r \in [d]$.

To estimate α and L in (11), we employ the idea in Gardes and Girard (2008). The quantile function corresponding to (11) is $q(p) := \inf\{s > 0 : \mathbb{P}(Y \le s) > p\} = L\left(\log \frac{2}{1-p}\right)^{\frac{1}{\alpha}}$. Taking the log on both sides, we have

$$\log q(p) = \frac{1}{\alpha}\log\log\frac{2}{1-p} + \log L, \quad p \in (0, 1),$$

which shows an affine relationship between $\log q(p)$ and $\log\log(2/(1-p))$ with the slope $1/\alpha$ and intersection $\log L$. So we can apply linear regression to estimate the slope $1/\alpha$ and the corresponding constant L using the tail observations. Specifically, assume that we have n i.i.d. observations ordered as $Y_r(1) \le Y_r(2) \le \cdots \le Y_r(n)$. Consider the largest k observations $Y_r(n-j)$ for $1 \le j \le k$. Each of these observations approximates the quantile $(n-j)/n = 1 - j/n$ (the largest observation $Y_r(n)$ corresponds to $q(1)$ and is not included). Thus, we use the slope from the linear regression of $\log Y_r(n-j)$ against $\log\log(2n/j)$, for $1 \le j \le k$, as an estimate of $1/\alpha$ and use the intersection as an estimate of $\log L$.

Note that of the two parameters, α is much important than L from a financial point of view, for α characterizes the heavy-tailedness of the return distribution. In contrast, L is only a multiplicative factor in Rule 1 which can be "absorbed" into the search interval $[a, b]$, namely, one can set $L = 1$ and search in the interval $[L^2 a, L^2 b]$ instead of $[a, b]$.

[6] The random variables in $\boldsymbol{\rho}^{-1/2}X^*$ are uncorrelated. If we further assume that they are independent, then the w leading to the heaviest tail should only select the one variable with the heaviest tail.

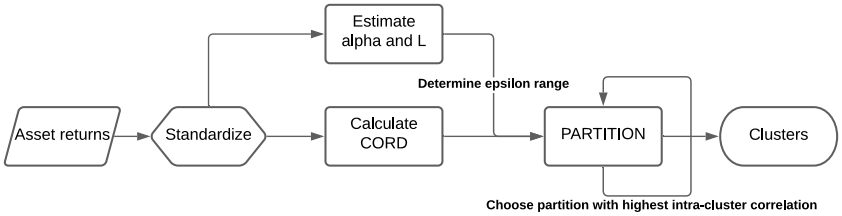


Fig. 1. Flowchart of the ACC algorithm.

2.5. The ACC algorithm

Summarizing the above, we now describe the complete algorithm that recovers clusters from the raw input $\boldsymbol{X}$, which we call the ACC (Asset Clustering through Correlation) algorithm; see Algorithm 1. We also illustrate it by a flowchart; see Fig. 1.

Algorithm 1 Asset Clustering through Correlation (ACC)

procedure ACC($\boldsymbol{X}$, a, b, n_g, $\mathcal{U}$, k) ▷ Input: returns $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, search range $[a, b]$, number of grids n_g, range of clusters $\mathcal{U}$, number of large observations k

$\boldsymbol{X}^* \longleftarrow \frac{\boldsymbol{X}-\text{mean}(\boldsymbol{X})}{\text{std}(\boldsymbol{X})}$ ▷ Standardized returns; mean and std are column-wise

$\widehat{\boldsymbol{\rho}} \longleftarrow \frac{1}{n-1}(\boldsymbol{X}^*)^\top \boldsymbol{X}^*$

$\widehat{\text{CORD}}(i,j) \longleftarrow \max_{l\neq i,j} |\widehat{\rho}_{il} - \widehat{\rho}_{jl}|, \quad \forall i,j \in [d]$

for $i \in [d]$ **do**

$Y_i \longleftarrow |(\boldsymbol{X}^*\widehat{\boldsymbol{\rho}}^{-1/2})_r| \in \mathbb{R}^n$

Sort Y_i such that $Y_i[1] \leq Y_i[2] \leq \ldots \leq Y_i[n]$

Slope s, intersection $a \longleftarrow$ LinearRegression($\log Y_i[n-k:n-1] \sim \log\log(2n/[1:k])$)

$\alpha_i \longleftarrow 1/s$, $L_i \longleftarrow \exp(a)$

$\alpha \longleftarrow \min_{i\in[d]} \alpha_i$

$L \longleftarrow \max_{i\in[d]} L_i$

if $n > (\log d)^{\frac{4}{\alpha}-1}$ **then**

Range $\mathcal{T} \longleftarrow [a,b] \times L^2\sqrt{\frac{\log d}{n}}$

else

Range $\mathcal{T} \longleftarrow [a,b] \times L^2 \frac{(\log d)^{2/\alpha}}{n}$

Uniformly divide range $\mathcal{T}$ into n_g grids

for ε in the n_g grids of $\mathcal{T}$ **do**

$\widehat{G}_\varepsilon \longleftarrow$ PARTITION($\widehat{\text{CORD}}, \varepsilon$)

if $|\widehat{G}_\varepsilon| \in \mathcal{U}$ **then** $\widehat{\rho}_\varepsilon^{ave} \longleftarrow \frac{\sum_{i<j} 1\left(i \overset{\widehat{G}_\varepsilon}{\sim} j\right)\widehat{\rho}_{ij}}{\sum_{i<j} 1\left(i \overset{\widehat{G}_\varepsilon}{\sim} j\right)}$

else

$\widehat{\rho}_\varepsilon^{ave} \longleftarrow -\infty$.

$\varepsilon_{\mathcal{T}} \longleftarrow \text{argmax}_{\varepsilon\in\mathcal{T}} \widehat{\rho}_\varepsilon^{ave}$

return PARTITION($\widehat{\text{CORD}}, \varepsilon_{\mathcal{T}}$)

The algorithmic complexity of ACC is polynomial in both the number of assets d and the number of observations n. Specifically, we have the following theorem.

Theorem 4. *The ACC algorithm requires at most $\mathcal{O}(nd^2 + d^3)$ arithmetic operations.*

We prove Theorem 4 in Appendix A.4.

3. Empirical analysis

This section reports the results of empirical experiments applying the ACC algorithm to financial time series data. Specifically, we cluster the stocks in the S&P 500 universe using our ACC algorithm, together with two benchmarks: the k-medoids algorithm and the single-linkage hierarchical clustering algorithm, and analyze the quality of such clustering results. Then based on the clustering results, we construct stock portfolios using three allocation strategies: the risk parity strategy, the minimum variance strategy, and Markowitz's mean–variance strategy, and compare the performances of these portfolios. We include additional benchmark portfolios based on the GICS sector and industry group classification and also portfolios of S&P 500 sector ETFs. These benchmarks will be explained in detail in Section 3.3.

3.1. Data preparation

We take the constituents of the S&P 500 as the universe. The data is obtained from Compustat through Wharton Research Data Services (WRDS), which consists of

- the daily closing prices of the constituents;
- the historical constituents data; and
- the daily closing S&P 500 total return index with dividends reinvested,

between January 1996 and January 2020.

We conduct clustering and backtesting for the period between February 2001 and January 2020. Clustering and portfolios are calculated on the first trading day of each month on the S&P 500 constituent stocks. Specifically, at the end of the first trading day of each month, we choose the stocks in the S&P 500 index according to the historical constituents data. Of all the current constituents, we discard stocks with less than five years of history and those with more than 5% missing data in the past $n = 500$ days. If the same company has multiple classes of stocks in the S&P 500 index (e.g., Alphabet Inc's GOOG and GOOGL), we only choose the class with the longest history. After the above filtering, the number of eligible stocks ranges between 465 and 488 over the backtesting period. For these eligible stocks, any missing prices are linearly interpolated using the previous and subsequent prices. Then, clusters are estimated based on the daily returns of the past $n = 500$ trading days. A smaller set of stocks is then selected, and portfolios are constructed using different allocation strategies. We defer the details of portfolio construction to Section 3.3.

3.2. Clustering procedure

We now give a detailed description of how we use the ACC algorithm. On the first trading day of each month, the ACC algorithm is applied to the sample correlation matrix in the backward 500-trading-day window for valid constituent stocks as described in Section 3.1. The following highlights more specifics:

- The heavy-tailedness parameter α and constant L are estimated by the approach in Section 2.4, where we choose $k := n/4 = 125$.
- The search range for the threshold parameter ε is set by Rule 1 with $a = 0.1$, $b = 10$ and $N = 100$. That is, if $n > \log(d)^{\frac{4}{\alpha}-1}$, then the range is $\mathcal{T} = [0.1, 10] \times L^2\sqrt{\frac{\log(d)}{n}}$; otherwise $\mathcal{T} = [0.1, 10] \times L^2\frac{\log(d)^{\frac{2}{\alpha}}}{n}$. We cap the upper bound of the search range to 2. The grid search is then performed on the search range with the range split into 100 grids.

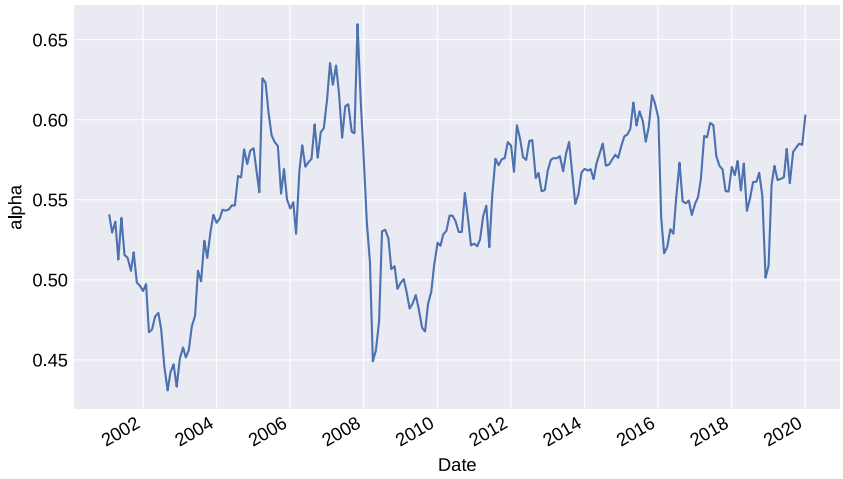


Fig. 2. Estimated α values by month via 500-trading-day windows.

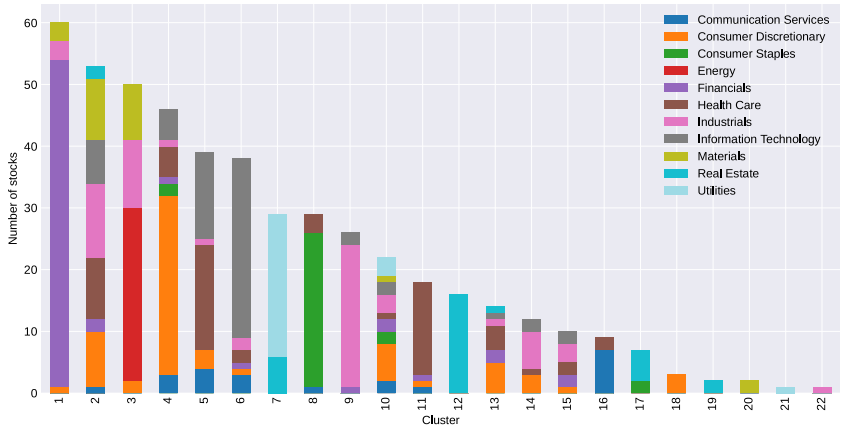


Fig. 3. ACC cluster compositions on 2019-02-01 compared with GICS sectors, choosing between 15 and 25 clusters.

- The number of clusters in Rule 3 is restricted between 15 and 25, i.e. $\mathcal{U}^* = [15, 25]$.

From February 2001 to January 2020, a total of 228 partitions are constructed, one in each month. The estimated α values estimated at the beginning of each month are plotted in Fig. 2. The S&P 500 data exhibits notable heavy tails, with α values lower than 1. The estimated value of the constant L ranges between 0.6 and 0.8 over time. It is also worth pointing out that the value of α drops significantly between 2008 and 2010, arguably due to more extreme returns observed during the 2008 financial crisis.[7]

To examine the compositions of the clusters, we compare the clusters with sectors defined by the Global Industry Classification Standard (GICS).[8] Fig. 3 shows the clustering result obtained on Feb 1st, 2019 from the above procedure. We observe that some of the clusters largely overlap with GICS sectors; e.g., Cluster 9 (Industrials, specifically Aerospace & Defense), 1 (Financials), 6 (Information Technology), 8 (Consumer Staples), 11 (Health Care), 12 (Real Estate). However, there are also ample discrepancies between these clusters and the GICS clusters. For instance, Cluster 6 consists of mostly stocks classified as information technology by GICS. A few notable stocks in that cluster not classified as information technology by GICS are Amazon.com (consumer discretionary), Alphabet, Netflix, Electronic Arts (communication services), S&P Global (Financials), and Rockwell Automation (Industrials). Upon closer examination, one may find that those companies are highly associated with the IT industry in their business nature. Indeed, Amazon.com, Netflix, and Alphabet are often perceived as IT companies and usually mentioned together as members of "FAANG".[9] In fact, four of FAANG are in this cluster, with the exception being Facebook. Electronic Arts, a software company that creates video games, is also naturally associated with the IT industry. S&P Global's primary business in financial information and analytics is likely why it is highly related to information technology, especially in today's world where finance is largely online and digital. Although Rockwell Automation, formerly Rockwell International, was known for manufacturing aircraft and electronic components, its current business lies in control systems and software applications for industrial automation. Another two companies in Cluster 6, yet not classified as information technology by GICS, are in the health care industry: Intuitive Surgical, which develops robotic surgical products, and Agilent Technologies, which provides analytical instruments and technology platforms for laboratories. In these examples, our clustering appears to gather technology companies in this cluster, not according to any existing taxonomy, but by discovering the associations of their business nature through the stock market movements. There is still, however, one exception that does not quite belong by the business

[7] This is consistent with the increased tail risk measured by VaR observed in stock daily returns during the financial crisis; see, e.g., Chaudhury (2014).

[8] Available at https://www.msci.com/gics.

[9] The FAANG includes Facebook, Amazon, Apple, Netflix, and Alphabet (Google).

Table 2
Performance of non-IT stocks in Cluster 6 between 2019-02-01 and 2020-02-01, compared with the FAANG stocks.

Ticker	Company name	Ann. Sharpe ratio	Ann. Return
FB	Facebook Inc	0.86	21.94%
AMZN	Amazon.Com Inc	1.12	23.62%
AAPL	Apple Inc	3.81	89.15%
NFLX	Netflix Inc	0.05	1.55%
GOOGL	Alphabet Inc	1.24	28.21%
EA	Electronic Arts Inc	0.52	18.39%
SPGI	S&P Global Inc	2.84	53.3%
ROK	Rockwell Automation	0.6	16.2%
ISRG	Intuitive Surgical Inc	0.26	6.72%
A	Agilent Technologies Inc	0.4	9.53%
FBHS	Fortune Brands Home & Secur	2.28	55.87%

nature: Fortune Brands Home & Security, which is a manufacturer of home fixtures and hardware.

The above example demonstrates a characterizing feature of the correlation blockmodel as opposed to traditional industry classifications: the ACC algorithm, which is purely data-driven, can identify stock groups based on correlation similarities rather than relying on fundamental information or knowledge about the companies' business. One advantage of this feature in asset selection and allocation is that we can uncover those "under-the-radar" stocks that can be used to replace the "big name" stocks (such as the FAANG) – the latter are often over-owned and hence tend to be over-priced – in a well-diversified portfolio.

Indeed, even if we just compare the *individual* performance of the six non-IT stocks in Cluster 6 with the FAANG in the one year *after* the clustering on February 1st, 2019, the results are noteworthy.Table 2 shows the results. In particular, Fortune Brands Home & Security and S&P Global have had decent annualized Sharpe ratios and returns compared to the IT stocks. This shows the potential of utilizing the clustering information to identify less-known stocks that have good performance and provide good diversification in a portfolio.

In addition to Cluster 6, which largely overlaps with a single GICS sector, we have Cluster 7, consisting of two closely related sectors: Real Estate and Utilities. There are also clusters that do not show any apparent theme that aligns with any particular GICS sector. For example, Clusters 2 and 10 both include stocks in all but a few GICS sectors. This fact reaffirms that the clustering is providing information that cannot be reflected in the industrial classification nor by mere common knowledge or experience.

Next, we compare ACC with two other benchmark clustering methods. The first is the single-linkage clustering, which is an instance of the classical hierarchical clustering method (Anderberg, 1973). The same method is used in Mantegna (1999) to analyze the hierarchical structure in the financial market. The second method is the k-medoids method (Kaufman & Rousseeuw, 1990), which is based on the search of k representative stocks as medoids and the assignment of every other stock to the closest medoid. These two clustering methods, briefly reviewed in Appendices B.1 and B.2, are two major unsupervised clustering approaches; see e.g. Hastie et al. (2009, Section 14.3). The hierarchical method belongs to *connectivity-based clustering* while k-medoids method to *centroid-based clustering*. Many newer clustering methods are variants of or follows the same spirit as these two methods; so taking them as benchmarks should be sufficiently representative. On the other hand, these two methods for asset clustering are purely heuristic, and they are both based on Criterion 1 *only*, namely to cluster in a way that assets in the same group have high correlations.

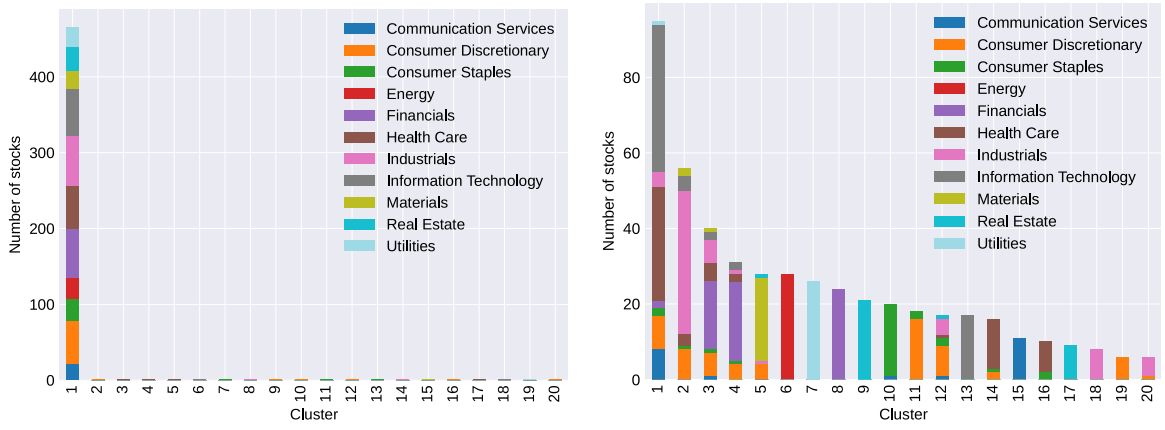
Fig. 4 shows their results obtained on Feb 1st, 2019. We observe in Fig. 4(a) that the clusters obtained from the hierarchical clustering are not very useful, with almost all stocks concentrating in one giant cluster, while the other clusters are mostly singletons. This finding is robust across all sliding windows that we tested. Results with similar characteristics are also reported in Musmeci et al. (2015). These results show that hierarchical clustering is very sensitive to the existence of global factors that load on a large number of stocks. When most or all stocks are sufficiently correlated with one another, those stocks will be merged with priority by hierarchical clustering and form an oversized cluster.

The clusters from k-medoids, as shown in Fig. 4(b), also overlap to some extent with the GICS sectors; yet the compositions are sufficiently different from our ACC results. For example, k-medoids puts Financial stocks mainly into three clusters: Cluster 3, Cluster 4, and Cluster 8, in each of which Financials are the majority, whereas ACC groups most Financial stocks together in Cluster 1 of Fig. 3. In the ACC partition, Consumer Discretionary stocks are more concentrated in a single cluster (Cluster 4), while in k-medoids clustering, they are more scattered across different clusters. In ACC, we have observed that four of the FAANG stocks are grouped in the same cluster, together with many other IT stocks and six stocks from other sectors. In k-medoids, all FAANG stocks are grouped together in Cluster 1, which is the largest cluster containing a majority of the IT stocks but also stocks in Utilities, Industrials, Health Care, Financials, Consumer Staples, Consumer Discretionary, and Communications Services. Visually, it appears that the k-medoids clusters are more similar to GICS sectors than ACC. To quantify this similarity, we calculate the adjusted Rand index R_{adj} (Hubert & Arabie, 1985) between the clustering and the GICS sectors. R_{adj} is an index that measures the similarity between two different partitions on the same set of objects, with the value 0 representing no similarity and the value 1 representing identical partitions. We calculate R_{adj} for both ACC and k-medoids clusterings compared to GICS sectors for each month. The results are presented in Fig. 5. Indeed, k-medoids clustering almost always produces more similar results to the GICS sectors than ACC does. Hence, ACC provides a more distinctive alternative clustering to the existing GICS classification. In particular, if one of the benefits of clustering stocks, as discussed earlier, is to technically unearth less known names whose price movements are similar to those of the big names, then the one producing results less similar to the GICS sectors would be advantageous.

3.3. Portfolio construction and backtesting

To construct portfolios based on the ACC clustering results, we first need to determine which stock(s) to select from each cluster. Theoretically, this selection is hinted by Theorem 2: if the return distribution follows exactly the correlation blockmodel, then one ought to select the stock(s) with the lowest volatility from each cluster. However, the real data do not follow exactly the correlation blockmodel; so empirically and practically it matters how one selects stock(s) in each cluster. There are two natural criteria for this selection: Sharpe ratio (i.e. risk-adjusted return) and volatility. In this paper, we choose the latter for the following reasons. First, low volatility as a criterion does not involve the estimation of the mean returns. ACC and the k-medoids algorithm tested do not cluster stocks based on their mean returns but only their correlations. So it would be inconsistent if we selected stocks from the clusters based on return-related criteria including Sharpe ratio. Second, although estimations of mean and variance based on historical data are both biased, it is well known that errors in estimated mean (a.k.a the "mean-blur" problem; see e.g. Merton, 1980) are of much greater significance than those in estimated variance. Thus, using volatility instead of Sharpe ratio (which has two source errors involving both estimated mean and variance) appears to be more reasonable. Finally, stocks with low volatility have been observed empirically to outperform the benchmarks over time, which is contrary to CAPM and is documented as the "low-risk anomaly" (e.g., Zaremba & Shemer, 2017).

In our experiments, the volatility is computed using the sample variance of daily returns in the past 500 trading days. We only choose one stock from each cluster, so the number of stocks in the portfolio



(a) Hierarchical clustering (b) k-medoids clustering

Fig. 4. Hierarchical and k-medoids cluster compositions on 2019-02-01 compared with GICS sectors.

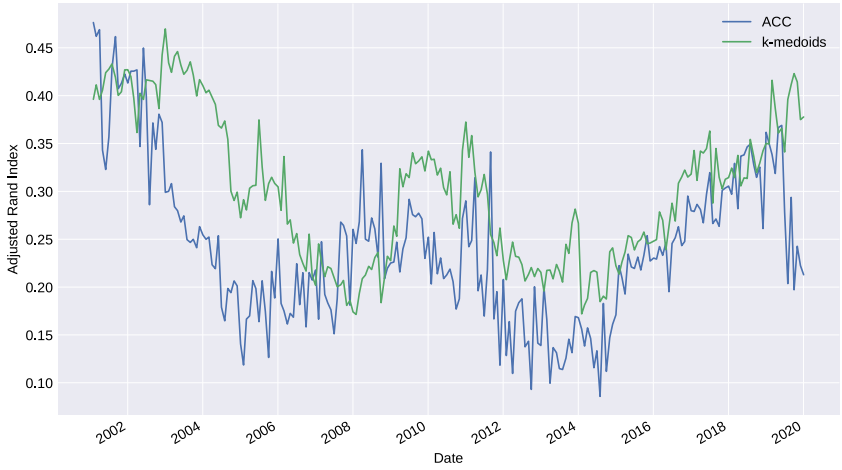


Fig. 5. The adjusted Rand index R_{adj} of ACC and k-medoids clusterings compared with GICS sectors.

equals the number of clusters discovered by the ACC algorithm. For comparison, we select stocks using the same criterion on the results of k-medoids clustering, where we select 20 stocks from 20 clusters, and also from GICS sectors and industry groups, where we select one stock from each sector and each industry group, respectively. The GICS sectors and industry groups are "point-in-time", meaning that stocks selected on a given date are based on their GICS classifications on that date in history. There are 10 GICS sectors before September 2016 and 11 afterward. Similarly, there are 22 GICS industry groups before December 2001, 23 before April 2003, and 24 afterward. These numbers determine the numbers of stocks selected in history based on the GICS sectors and industry groups.

Once a set of stocks is determined by the above procedure, three asset allocation strategies are employed and compared. The first strategy is the risk parity strategy, which, since the 2008 crisis, has become one of the most popular approaches among portfolio managers. The second strategy is the minimum variance allocation strategy. The third strategy is Markowitz's mean–variance strategy without short-selling, where we set the target annualized return to 10%. We briefly review these three allocation strategies in Appendices B.3 and B.4.

Recall that for ACC, the number of the clusters in Rule 3 is set between 15 and 25, yielding 15 to 25 stocks. The exact number of clusters discovered by ACC over time, reported in Fig. 6, fluctuates wildly, and hence the resulting clusters are likely very different between months. If we were to construct a portfolio using the clustering results every month, we would be in and out of positions very frequently as the stocks in the portfolio will likely be changing from month to month, which is practically not desirable, especially considering transaction costs and tax implications. Therefore, we will only readjust the portfolios once every year. Specifically, for each of the clustering methods, on the first trading day of each February, a new set of stocks is selected according to the clustering result, and their allocations under the three aforementioned strategies are respectively calculated using all daily returns in the past 500 trading days, starting with the first day when all stocks are available. The positions are then held until the first trading day of the following February. Any dividends are immediately reinvested in the same stock. We assume no transaction cost for simplicity.

As benchmarks, we first take the S&P 500 ETF (NYSE ticker: SPY), which is the world's largest ETF and is designed to track the S&P 500 stock market index. We also create portfolios consisting of all S&P 500 sector ETFs[10] using the above three allocation strategies, all rebalanced annually. Each of these ETFs consists of companies in a specific GICS sector in the S&P 500 Index. Investing in these ETFs represents a simple method of diversifying among sectors while maintaining the ability to decide the weight of each sector optimally. Finally, as an "extreme"

[10] https://www.ssga.com/us/en/individual/etfs/capabilities/invest-with-sector-etfs/sector-and-industry-etfs.

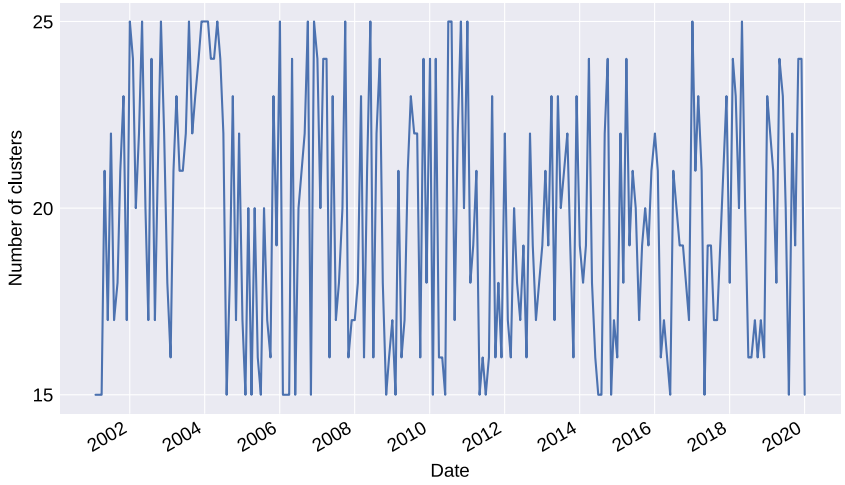


Fig. 6. Number of clusters discovered by ACC over time.

Table 3
Performance metrics of the risk parity portfolios. ACC chooses between 15 and 25 stocks, and k-medoids chooses 20 stocks. Rebalanced annually.

	ACC	GICS sector	GICS ind. group	k-medoids	Sector ETFs	All stocks	SPY
Ending VAMI	8206.92	6439.89	5931.68	6305.41	4334.0	5874.65	3442.53
Max Drawdown	44.08%	36.89%	42.6%	47.26%	49.56%	54.69%	55.25%
Peak-To-Valley	2007-06-01–2009-03-05	2007-12-10–2009-03-09	2007-10-12–2009-03-09	2007-06-04–2009-03-09	2007-10-12–2009-03-09	2007-07-13–2009-03-09	2007-10-09–2009-03-09
Recovery	261 Days	445 Days	485 Days	445 Days	492 Days	467 Days	774 Days
Sharpe Ratio	0.79	0.74	0.68	0.7	0.49	0.56	0.36
Sortino Ratio	1.29	1.2	1.09	1.14	0.77	0.89	0.57
Calmar Ratio	0.27	0.28	0.23	0.22	0.16	0.18	0.12
Ann. Volatility	14.83%	13.98%	14.48%	14.48%	16.34%	17.41%	18.63%
Ann. Downside Volatility	9.1%	8.63%	9.04%	8.95%	10.39%	11.04%	11.83%
Correlation	0.9	0.89	0.94	0.93	0.98	0.98	1.0
Beta	0.72	0.67	0.73	0.72	0.86	0.92	1.0
Ann. Return	11.75%	10.33%	9.85%	10.21%	8.05%	9.8%	6.74%
Ann. Turnover Ratio	80.59%	51.03%	42.91%	67.09%	6.01%	14.52%	–
Positive Periods	2631 (55.11%)	2554 (53.50%)	2568 (53.79%)	2575 (53.94%)	2626 (55.01%)	2613 (54.73%)	2600 (54.46%)
Negative Periods	2143 (44.89%)	2220 (46.50%)	2206 (46.21%)	2199 (46.06%)	2148 (44.99%)	2161 (45.27%)	2174 (45.54%)

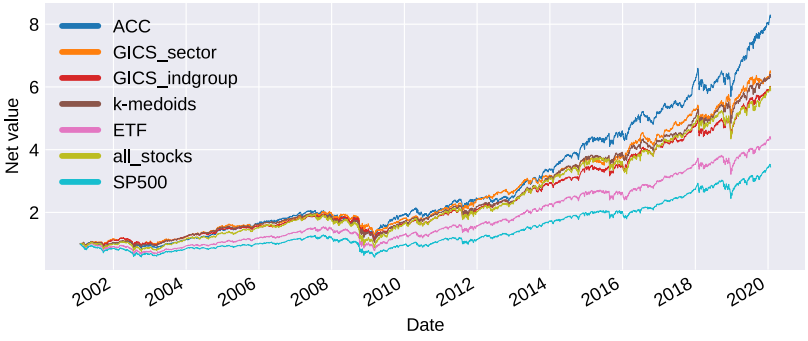
benchmark, we include portfolios consisting of *all* eligible stocks (after filtering out those with missing data and insufficient history) in our comparisons with the three annually rebalanced allocation strategies.

Fig. 7 shows the daily values of these portfolios along with SPY, and Tables 3–5 report results based on performance metrics commonly used in the wealth management industry. ACC outperforms SPY significantly in the most important return and risk metrics (including the Sharpe, Sortino, and Calmar ratios, annual volatility, and annualized return) under all three strategies. ACC also has a much smaller maximum drawdown than SPY. It also significantly outperforms the portfolios of sector ETFs under all allocation strategies. This observation shows the advantage of asset selection using clustering over simply constructing portfolios using the sector ETFs. Between ACC and the portfolios based on GICS classifications, ACC portfolios offer better return–risk ratios and faster recovery from the maximum drawdown. The "all-stock" portfolios, on the other hand, do indeed beat SPY significantly, but they still generally underperform the corresponding ACC portfolios while the underperformance is substantial with the risk parity strategy. Between ACC and k-medoids, ACC is also superior when employing the risk parity and the mean–variance allocation strategies. With the minimum-variance strategy, ACC still offers a better annualized return, a smaller maximum drawdown, and faster recovery than k-medoids while maintaining a similar overall Sharpe ratio. Moreover, as discussed earlier, ACC has the advantage of having a theoretical foundation, whereas k-medoids is a pure heuristic for financial time series.

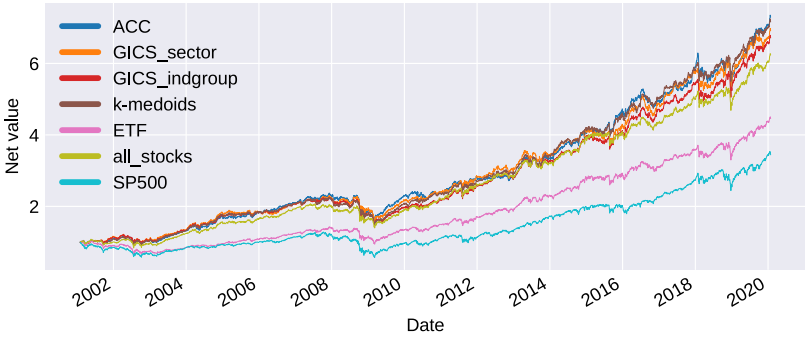
The above results compare the overall performance of the portfolios throughout the entire 19-year period. For a more comprehensive comparison, we break down the 19-year periods into small sub-periods using a rolling window approach and compare the returns among all sub-periods. We first set the length of the rolling window to be one calendar year and capture the returns of the portfolios within all 1-year sub-periods, e.g., from 2001-02-01 to 2002-02-01 and from 2001-02-02 to 2002-02-02. We compare the annualized return and the annualized Sharpe ratio between the ACC portfolios and the benchmarks in each sub-period. Then we repeat the analysis for different lengths of the rolling window. Fig. 8 shows the percentage of sub-periods of different lengths where ACC has a higher return than the benchmarks, with the three allocation strategies, respectively. Fig. 9 reports the same comparison for Sharpe ratios.[11]

As the window length approaches ten years and longer, all three portfolios from ACC clustering almost *always* outperform SPY, in both return and Sharpe ratio. This observation shows that the ACC portfolios are very suitable for investors with long investment horizons. Even for investors with short investment horizons like 1 to 2 years, ACC is still more likely to achieve better returns and Sharpe ratios than SPY. Compared with the other benchmarks, ACC is superior in the long run to all but the k-medoids portfolio with the minimum variance

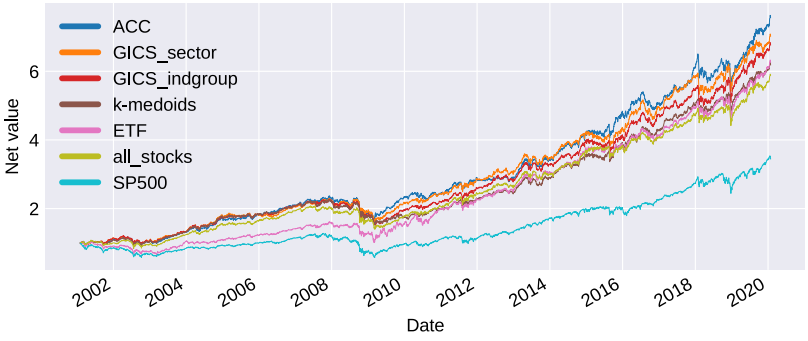
[11] As comparing negative Sharpe ratios is meaningless, we exclude windows in which both ACC and the benchmark have negative Sharpe ratios.



(a) Risk parity portfolios



(b) Minimum variance portfolios



(c) Single-period mean-variance portfolios with 10% target return

Fig. 7. Daily value comparison among the portfolios. ACC chooses between 15 and 25 stocks, and k-medoids chooses 20 stocks. Rebalanced annually.

allocation strategy. ACC seems to underperform the portfolio of sector ETFs in annualized returns for a large range of window lengths when employing the mean–variance strategy. Still, ACC's Sharpe ratio tends to be superior in the long run compared to the sector ETF portfolio.

In addition to rebalancing the portfolios once every year (i.e., annual rebalancing), we also test semi-annual and quarterly rebalancing (i.e., every 6 and 3 months, respectively). Table 6 reports the results. ACC appears to lose some advantage when the portfolios are rebalanced more frequently. This is consistent with the previous observation that ACC clustering changes frequently, and hence, intuitively, portfolios updated frequently based on the ACC clusters might not perform well. Still, ACC portfolios consistently outperform the SPY and the portfolios of sector ETFs. Furthermore, with annual rebalancing, ACC not only achieves the highest Sharpe ratio with the risk parity and the mean–variance strategies among all annually rebalanced portfolios, but its Sharpe ratios are also the highest among all portfolios with the same allocation strategies regardless of rebalancing frequencies. With the minimum variance allocation, ACC with annual rebalancing ranks third among all minimum variance portfolios with different rebalancing frequencies and only slightly lags behind the top two. Overall, we can conclude that ACC is a consistent and robust performer with different rebalancing frequencies and strategies, and its performance stands out for slow portfolios that do not require frequent rebalancing. It is particularly suitable for investors/funds whose investment philosophy is for less trading, if not completely "buy and hold". This characteristic of ACC has practical significance, since rebalancing more frequently than annually has unfavorable tax implications, and indeed investment experts have found no significant advantage of rebalancing portfolios more frequently once transaction costs and taxes are taken into consideration (McNamee et al., 2019; Zilbering et al., 2015).

We also observe from all these experiments that the outperformance of ACC when combined with the risk-parity allocation is overall more significant than when combined with the mean–variance or minimum variance allocations. This may be intuitively explained as follows. Numerical experiments (Maillard et al., 2010) show that risk parity often provides more balanced allocations, mitigating the problem of

Table 4
Performance metrics of the minimum variance portfolios. ACC chooses between 15 and 25 stocks, and k-medoids chooses 20 stocks. Rebalanced annually.

	ACC	GICS sector	GICS ind. group	k-medoids	Sector ETFs	All stocks	SPY
Ending VAMI	7299.02	6912.71	6715.02	7159.2	4455.71	6239.42	3442.53
Max Drawdown	32.45%	30.56%	34.04%	34.06%	37.65%	32.64%	55.25%
Peak-To-Valley	2007-12-10–2009-03-09	2007-12-10–2009-03-09	2007-12-10–2009-03-09	2007-12-10–2009-03-09	2001-05-21–2002-07-23	2007-06-04–2009-03-09	2007-10-09–2009-03-09
Recovery	250 Days	380 Days	516 Days	377 Days	747 Days	393 Days	774 Days
Sharpe Ratio	0.84	0.81	0.82	0.85	0.61	0.85	0.36
Sortino Ratio	1.38	1.35	1.34	1.4	0.98	1.38	0.57
Calmar Ratio	0.34	0.35	0.31	0.32	0.22	0.31	0.12
Ann. Volatility	13.19%	13.21%	12.95%	12.81%	13.36%	11.97%	18.63%
Ann. Downside Volatility	8.0%	7.97%	7.9%	7.82%	8.39%	7.35%	11.83%
Correlation	0.78	0.77	0.8	0.79	0.85	0.83	1.0
Beta	0.55	0.55	0.56	0.54	0.61	0.53	1.0
Ann. Return	11.06%	10.74%	10.57%	10.95%	8.21%	10.15%	6.74%
Ann. Turnover Ratio	77.78%	56.17%	58.56%	62.22%	20.78%	64.31%	–
Positive Periods	2597 (54.40%)	2567 (53.77%)	2580 (54.04%)	2579 (54.02%)	2600 (54.46%)	2614 (54.75%)	2600 (54.46%)
Negative Periods	2177 (45.60%)	2207 (46.23%)	2194 (45.96%)	2195 (45.98%)	2174 (45.54%)	2160 (45.25%)	2174 (45.54%)

Table 5
Performance metrics of the single period mean–variance portfolios. ACC chooses between 15 and 25 stocks, and k-medoids chooses 20 stocks. Rebalanced annually.

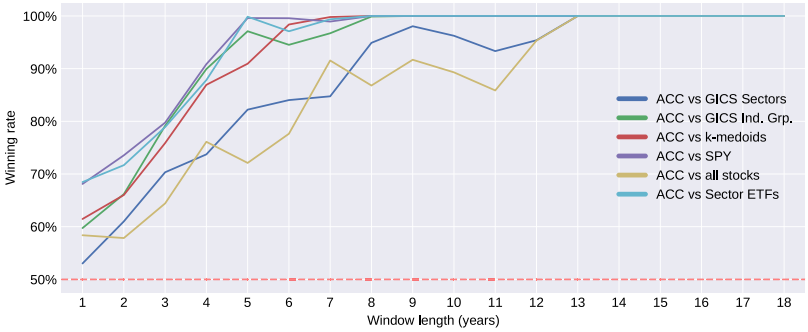
	ACC	GICS sector	GICS ind. group	k-medoids	Sector ETFs	All stocks	SPY
Ending VAMI	7575.26	7019.87	6762.27	6180.35	6257.01	5885.41	3442.53
Max Drawdown	31.16%	29.51%	33.21%	32.64%	37.55%	32.7%	55.25%
Peak-To-Valley	2007-12-10–2009-03-05	2007-12-10–2009-03-11	2007-12-10–2009-03-11	2007-12-10–2009-03-05	2007-12-10–2009-03-02	2007-06-04–2009-03-11	2007-10-09–2009-03-09
Recovery	241 Days	374 Days	517 Days	600 Days	215 Days	448 Days	774 Days
Sharpe Ratio	0.86	0.82	0.82	0.77	0.64	0.82	0.36
Sortino Ratio	1.42	1.36	1.34	1.26	1.03	1.33	0.57
Calmar Ratio	0.36	0.37	0.32	0.31	0.27	0.3	0.12
Ann. Volatility	13.18%	13.27%	12.99%	13.13%	15.8%	11.98%	18.63%
Ann. Downside Volatility	7.97%	8.0%	7.92%	8.03%	9.85%	7.36%	11.83%
Correlation	0.76	0.77	0.79	0.8	0.84	0.82	1.0
Beta	0.54	0.55	0.55	0.56	0.72	0.53	1.0
Ann. Return	11.28%	10.83%	10.62%	10.09%	10.16%	9.81%	6.74%
Ann. Turnover Ratio	78.07%	59.01%	60.41%	65.16%	51.14%	66.44%	–
Positive Periods	2585 (54.15%)	2557 (53.56%)	2591 (54.27%)	2552 (53.46%)	2590 (54.25%)	2618 (54.84%)	2600 (54.46%)
Negative Periods	2189 (45.85%)	2217 (46.44%)	2183 (45.73%)	2222 (46.54%)	2184 (45.75%)	2156 (45.16%)	2174 (45.54%)

Table 6
Sharpe ratio comparisons with different rebalancing frequencies.

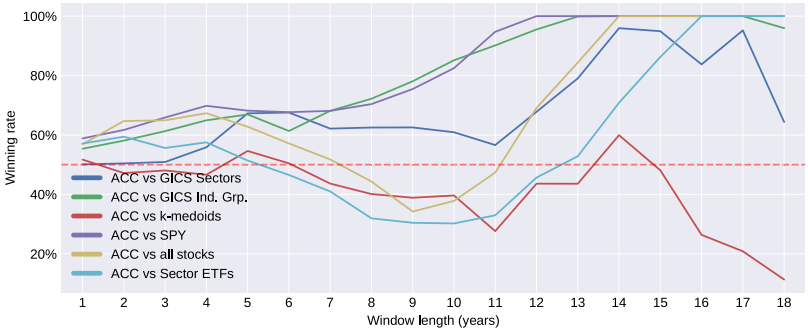
(a) Annual rebalancing							
	ACC	GICS Sector	GICS Ind. Grp.	k-medoids	SPY	All stocks	Sector ETF
risk parity	0.79	0.74	0.68	0.7	0.36	0.58	0.49
min-variance	0.84	0.81	0.82	0.85	0.36	0.85	0.61
mean variance	0.86	0.82	0.82	0.77	0.36	0.82	0.64
(b) Semi-annual rebalancing							
	ACC	GICS Sector	GICS Ind. Grp.	k-medoids	SPY	All stocks	Sector ETF
risk parity	0.72	0.79	0.69	0.66	0.36	0.54	0.48
min-variance	0.79	0.86	0.82	0.75	0.36	0.87	0.59
mean variance	0.78	0.83	0.81	0.71	0.36	0.84	0.56
(c) Quarterly rebalancing							
	ACC	GICS Sector	GICS Ind. Grp.	k-medoids	SPY	All stocks	Sector ETF
risk parity	0.72	0.76	0.69	0.65	0.36	0.54	0.48
min-variance	0.81	0.83	0.82	0.83	0.36	0.89	0.59
mean variance	0.78	0.78	0.81	0.8	0.36	0.86	0.5

extreme portfolios in the mean–variance approach (including minimum variance) while providing better returns–risk tradeoff than equally-weighted portfolios. It is therefore natural to expect that combining two better approaches (i.e. ACC and risk parity) would produce better results, as our experiments show empirically. However, we do not yet have a formal theory to explain why combining ACC with risk parity should be better than other possible combinations, and consider it a challenging open question that warrants a full study.

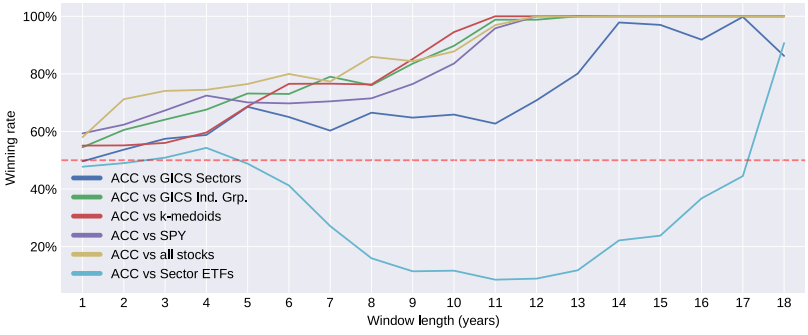
Seeing that the ACC portfolio consistently outperforms the benchmark SPY, it is intriguing to test a market-neutral portfolio that only captures the difference between the ACC portfolio and SPY. We construct a market-neutral portfolio using each of the annually rebalanced ACC portfolios that employ the three different allocation strategies.



(a) Risk parity



(b) Minimum variance



(c) Mean-variance

Fig. 8. Percentage of sub-periods in which ACC has a higher return than benchmarks under different allocation strategies.

At each rebalancing point (the end of the first trading day of each February) and for each stock i in the portfolio, we calculate its beta by linear regression of its return against the market return:

$$X_i = \alpha_i + \beta_i R_m + \varepsilon.$$

The beta has a closed-form solution,

$$\beta_i = \frac{\mathrm{Cov}(X_i, R_m)}{\mathrm{Var}(R_m)},$$

and is estimated using the sample covariance between stock returns and SPY returns and the sample variance of SPY returns, both of the past 500 trading days. Then, given a set of weights $\mathbf{w} = (w_1, w_2, \ldots, w_d)$ determined by one of the three allocation strategies, we calculate the total beta of the portfolio:

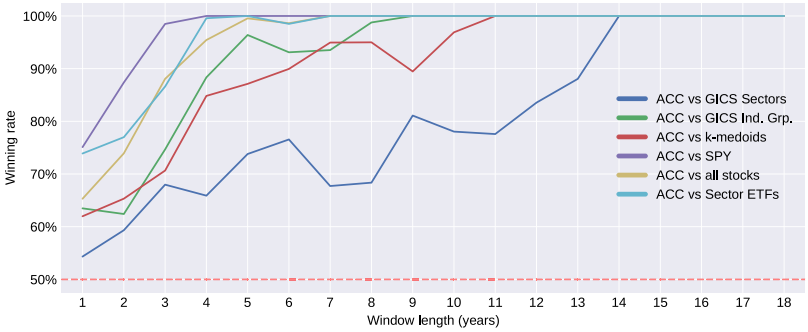
$$\beta = \sum_i^d w_i \beta_i.$$

Then, if we add a short position on SPY of weight $-\beta$, the portfolio will have zero exposure to SPY and hence become market neutral. However, now the portfolio has a leverage ratio of $1 + \beta$, so we scale the position down by dividing all positions by $1 + \beta$. The final weights of the market-neutral portfolio are

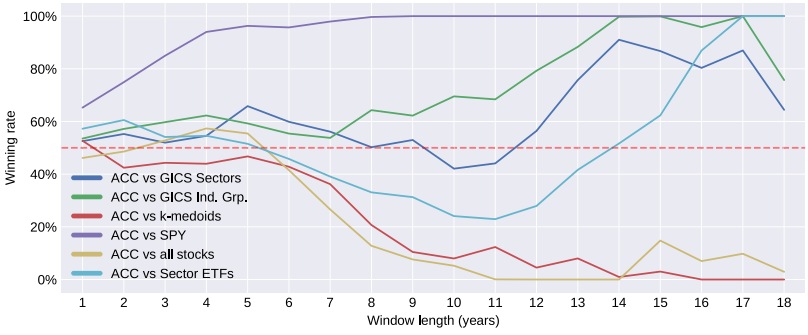
$$\left(\frac{w_1}{1+\beta}, \cdots, \frac{w_d}{1+\beta}, -\frac{\beta}{1+\beta}\right),$$

corresponding to the d stocks in the portfolio and the SPY. Fig. 10 and Table 7 show the net values and the performance metrics of the market-neutral portfolios based on the ACC clusters. Though the market-neutral portfolios have lower annualized returns, the risk parity market-neutral portfolio has a much higher Sharpe ratio than the original risk parity portfolio. This provides an alternative application of the ACC clustering results. If we zoom in on the period of the financial crisis (Dec 2007–Jul 2009), we can see in Table 8 that the market-neutral portfolios are very resilient to market downturns. All three portfolios obtain positive returns and have much lower drawdowns than the benchmark SPY.

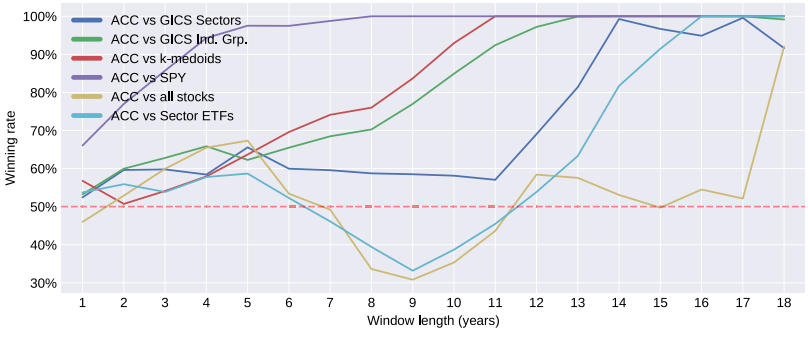
We have also repeated the same experiment for the constituents of the Russell 2000 Index, which, in contrast to S&P 500, consists of stocks with smaller market-caps. The outperformance of our ACC algorithm is more prominent. The results are reported in Appendix C.



(a) Risk parity



(b) Minimum variance



(c) Mean-variance

Fig. 9. Percentage of sub-periods in which ACC has a higher Sharpe ratio than benchmarks under different allocation strategies.

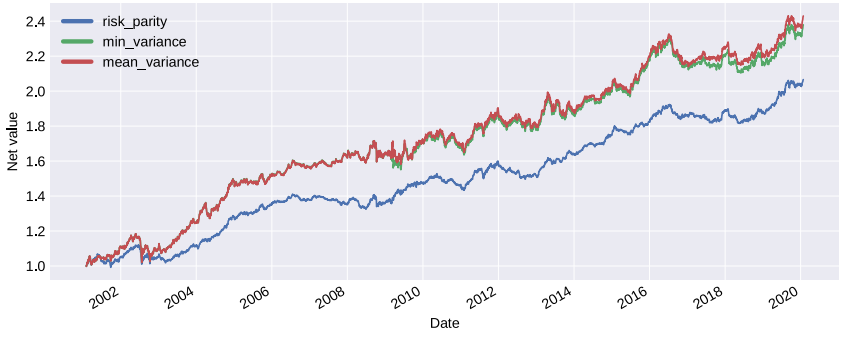


Fig. 10. Performance of ACC market neutral portfolios.

4. Conclusion

This paper aims to identify a smaller set of stocks that attains an adequate level of diversification compared to the whole universe of stocks. We achieve this by clustering financial assets via exploring the correlation structure. We cluster the assets in a group according to the joint correlation with all other assets. The idea is formalized by the correlation blockmodel, and the ACC algorithm is devised to cluster the model. We provide rigorous analysis of the ACC algorithm and give practical guidance based on the theoretical results. Numerical

Table 7
Performance of ACC market neutral portfolios.

	Risk parity	Min. variance	Mean–variance
Ending VAMI	2066.03	2379.17	2429.93
Max Drawdown	9.91%	13.78%	13.79%
Peak-To-Valley	2002-06-20–2002-07-23	2002-05-23–2002-07-23	2002-05-23–2002-07-23
Recovery	345 Days	254 Days	257 Days
Sharpe Ratio	0.93	0.77	0.77
Sortino Ratio	1.58	1.27	1.27
Calmar Ratio	0.39	0.34	0.35
Ann. Volatility	4.21%	6.05%	6.26%
Ann. Downside Volatility	2.47%	3.68%	3.78%
Correlation	0.3	0.26	0.22
Beta	0.07	0.08	0.07
Annualized Return	3.9%	4.68%	4.8%
Annualized Turnover Ratio	50.91%	54.42%	54.62%
Positive Periods	2522 (52.83%)	2508 (52.53%)	2505 (52.47%)
Negative Periods	2252 (47.17%)	2266 (47.47%)	2269 (47.53%)

Table 8
Performance of ACC market neutral portfolios between December 2007 and July 2009.

	Risk parity	Min. variance	Mean–variance	SPY
Ending VAMI	1056.22	1015.46	1023.28	652.15
Max Drawdown	4.85%	9.68%	7.81%	53.96%
Peak-To-Valley	2008-09-15–2008-10-10	2008-09-18–2009-06-01	2008-09-18–2009-06-01	2007-12-10–2009-03-09
Recovery	123 Days	–	–	–
Sharpe Ratio	0.67	0.11	0.14	−0.62
Sortino Ratio	1.17	0.18	0.24	−1.0
Calmar Ratio	0.73	0.1	0.19	−0.44
Ann. Volatility	5.25%	9.13%	10.38%	38.22%
Ann. Downside Volatility	3.03%	5.52%	6.24%	23.8%
Correlation	0.27	0.27	0.14	1.0
Beta	0.04	0.06	0.04	1.0
Annualized Return	3.53%	0.98%	1.47%	−23.77%
Annualized Turnover Ratio	49.69%	60.41%	60.15%	–
Positive Periods	197 (49.62%)	188 (47.36%)	185 (46.60%)	204 (51.39%)
Negative Periods	200 (50.38%)	209 (52.64%)	212 (53.40%)	193 (48.61%)

experiments show that portfolios constructed based on the ACC clustering algorithm achieve good performance compared to the market benchmark and also other portfolios consisting of similar numbers of assets.

Our work can be extended in several directions. One is to further improve the ACC algorithm. For instance, how to recover the clusters of the correlation blockmodel when the minimal separation $\min_{i \overset{G^\star}{\sim} j} \mathrm{CORD}(i,j)$ is below $\max(\sqrt{\log d\,/n}, (\log d)^{2/\alpha}/n)$? The other is to make use of the past return information to embed Criteria 1 & 2 directly into mean–variance optimization algorithms.

CRediT authorship contribution statement

Wenpin Tang: Conceptualization, Methodology, Implementation, Writing – original draft, Writing – review & editing. **Xiao Xu:** Conceptualization, Methodology, Implementation, Writing – original draft, Writing – review & editing. **Xun Yu Zhou:** Conceptualization, Methodology, Implementation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Tang gratefully acknowledges financial supports through an NSF, United States grant DMS-2113779 and through a start-up grant at Columbia University. Zhou gratefully acknowledges financial supports through a start-up grant at Columbia University and through the Nie Center for Intelligent Asset Management. We thank Kaizheng Wang for bringing the paper Bunea et al. (2020) to our attention. We are also indebted to two anonymous referees and the associate editor for detailed and constructive comments, which have led to an improved version of the paper.

Appendix A. Proofs of theoretical results

In this section, we prove the main results—Theorems 1–4.

A.1. Proof of Theorem 1

Assume that $\boldsymbol{\rho} = \boldsymbol{Z\Pi Z}^\top + \boldsymbol{\Gamma}$ holds for a membership matrix $\boldsymbol{Z}$ associated with some partition $G = \{G_1, G_2, \dots\}$. Then

$$\max_{l \neq i,j} |\rho_{il} - \rho_{jl}| = 0 \quad \text{for any } i, j \in G_k.$$

So each group G_k of G is included in one of the equivalence classes $\overset{G^\star}{\sim}$ defined by (3). As a result, the partition G is finer than $G^\star$. This implies that $G^\star$ is the unique coarsest partition such that the decomposition $\boldsymbol{\rho} = \boldsymbol{Z\Pi Z}^\top + \boldsymbol{\Gamma}$ holds.

A.2. Proof of Theorem 2

We first notice that for any portfolio P_J chosen by selecting one asset from each cluster in the coarsest partition G^*, the portfolio correlation matrix will be the same due to the block structure in the large correlation matrix among all assets. Denote this portfolio correlation matrix by $\boldsymbol{\rho}_P$. Then the portfolio covariance matrix is

$$\boldsymbol{\Sigma}_{P_J} = \boldsymbol{V}_{P_J} \boldsymbol{\rho}_P \boldsymbol{V}_{P_J},$$

where $\boldsymbol{V}_{P_J} := \operatorname{diag}\left(\sqrt{\operatorname{Var}(X_{J(1)})}, \ldots, \sqrt{\operatorname{Var}(X_{J(K)})}\right)$ is the diagonal matrix containing the standard deviation of all stocks in the portfolio. Because we do not allow short selling in the minimum variance portfolio, the optimal weights w^* fall into either of the following two cases.

1. All weights in w^* are positive, and $w^* = \boldsymbol{\Sigma}_{P_J}^{-1}1 / \left(1^\top \boldsymbol{\Sigma}_{P_J}^{-1}1\right)$.
2. Some weights in w^* are zero, and the remaining weights are positive and satisfy $w^{*+} = (\boldsymbol{\Sigma}_{P_J}^{+})^{-1}1 / \left(1^\top (\boldsymbol{\Sigma}_{P_J}^{+})^{-1}1\right)$, where $\boldsymbol{\Sigma}_{P_J}^{+}$ represents the covariance matrix of stocks with positive weights in the optimal portfolio.

We now show that, in both cases, the variance of the minimum variance portfolio is non-decreasing in all elements of $\boldsymbol{V}_{P_J}$.

In Case (1), where $w^* = \boldsymbol{\Sigma}_{P_J}^{-1}1 / \left(1^\top \boldsymbol{\Sigma}_{P_J}^{-1}1\right)$, the variance of the portfolio is

$$\operatorname{Var}_{\min}(P_J) = \frac{1}{21^\top \boldsymbol{\Sigma}_{P_J}^{-1}1} = \frac{1}{21^\top \boldsymbol{V}_{P_J}^{-1}\boldsymbol{\rho}_P^{-1}\boldsymbol{V}_{P_J}^{-1}1}. \tag{12}$$

Let us disregard the $1/2$ scaling and focus on the denominator, which can be rewritten as

$$\sum_{i=1}^{K}\sum_{j=1}^{K} \rho_P^{-1}(i,j)a_i a_j,$$

where $\rho_P^{-1}(i,j)$ is the jth element of the ith row of $\boldsymbol{\rho}_P^{-1}$, and $a_i := 1/\sqrt{\operatorname{Var}(X_{J(i)})}$. The derivative w.r.t. a_i is

$$2\rho_P^{-1}(i,i)a_i + 2\sum_{j\neq i} \rho_P^{-1}(i,j)a_j. \tag{13}$$

Because $w^* = \boldsymbol{\Sigma}_{P_J}^{-1}1 / \left(1^\top \boldsymbol{\Sigma}_{P_J}^{-1}1\right) = \boldsymbol{V}_{P_J}^{-1}\boldsymbol{\rho}_P^{-1}\boldsymbol{V}_{P_J}^{-1}1 / \left(1^\top \boldsymbol{\Sigma}_{P_J}^{-1}1\right) \geq 0$, we have for any $i \in [K]$,

$$\rho_P^{-1}(i,i)a_i^2 + \sum_{j\neq i} \rho_P^{-1}(i,j)a_i a_j \geq 0.$$

Dividing both sides by $a_i > 0$, we have

$$\rho_P^{-1}(i,i)a_i + \sum_{j\neq i} \rho_P^{-1}(i,j)a_j \geq 0.$$

This means that the derivative (13) is non-negative. In other words, the denominator of (12) is non-decreasing in a_i for any $i \in [K]$, thus (12) is non-decreasing in $\operatorname{Var}(X_{J(i)})$ for any $i \in [K]$.

Now let us consider Case (2). If the optimal weight for a stock X_i is positive, then, as we have shown in case (1), increasing $\operatorname{Var}(X_i)$ will decrease a_i and thus increase the variance of the optimal portfolio. This also decreases its optimal weight w_i^{*+}, up to the point where $w_i^{*+} = 0$. If the optimal weight for a stock X_i is zero, it means that, in the optimal minimum variance portfolio where short selling is allowed, the weight of X_i is non-positive. In other words,

$$\rho_P^{-1}(i,i)a_i^2 + \sum_{j\neq i} \rho_P^{-1}(i,j)a_i a_j \leq 0. \tag{14}$$

Notice that the left-hand side of (14) is quadratic in a_i, with one root at $a_i = 0$ and another at some $a_i > 0$. Because the $\rho_P^{-1}(i,i)$, being the diagonal of the inverse correlation matrix, is larger than 1, increasing the variance of X_i thus decreasing a_i toward 0 will only make (14) stay negative. Hence the optimal weight of the no-short-selling problem will not change. This shows that the variance of the minimum-variance portfolio is also non-decreasing in $\operatorname{Var}(X_{J(i)})$ for any $i \in [K]$ in case (2).

A.3. Proof of Theorem 3

We start with the following lemma, which provides a sufficient condition under which the PARTITION algorithm recovers the partition $G^\star$.

Lemma 1. *Let* $\tau := \max_{i,j,l\in[d]} |(\widehat{\rho}_{il} - \widehat{\rho}_{jl}) - (\rho_{il} - \rho_{jl})|$, *and* $\Delta := \min_{i \overset{G^\star}{\sim} j} \operatorname{CORD}(i,j)$. *If* $\tau \leq \varepsilon < \Delta - \tau$, *then the PARTITION algorithm with inputs* $\widehat{\operatorname{CORD}}$ *and* ε *outputs* $\widehat{G} = G^\star$.

Proof of Lemma 1. By the definition of τ, we have $\widehat{\operatorname{CORD}}(i,j) - \tau \leq \operatorname{CORD}(i,j) \leq \widehat{\operatorname{CORD}}(i,j) + \tau$. If $i \overset{G^\star}{\sim} j$, we have $\operatorname{CORD}(i,j) = 0$, and thus $\widehat{\operatorname{CORD}}(i,j) \leq \tau$. Similarly, if $i \overset{G^\star}{\nsim} j$, we have $\operatorname{CORD}(i,j) \geq \Delta$, and thus $\widehat{\operatorname{CORD}}(i,j) \geq \Delta - \tau$. Consequently, if $\tau \leq \varepsilon < \Delta - \tau$, then

$$i \overset{G^\star}{\sim} j \quad \text{if and only if} \quad \widehat{\operatorname{CORD}}(i,j) \leq \varepsilon.$$

Now we prove that the PARTITION algorithm recovers the partition $G^\star$. We argue by induction. Assume that the algorithm is correct in the first $t-1$ steps:

$$\widehat{G}_s = G^\star_{k(i_l)} \quad \text{for } l = 1, \ldots, t-1,$$

where $k(i_l)$ is the index of the group that contains i_l. At step t, if $|S| = 1$, the algorithm terminates after this step and outputs $\widehat{G} = G^\star$. If $|S| > 1$, there are two cases:

1. If $\widehat{\operatorname{CORD}}(i_t, j_t) > \varepsilon$, then $\widehat{\operatorname{CORD}}(i_t, j) > \varepsilon$ and thus $i_t \overset{G^\star}{\nsim} j$ for any $j \in S$. Since the algorithm is correct up to step $t-1$, $i_t \overset{G^\star}{\nsim} j$ for any $j \notin S$. Thus, i_t must be a singleton, and the algorithm outputs $\widehat{G}_t = G^\star_{k(i_t)} = \{i_t\}$.
2. If $\widehat{\operatorname{CORD}}(i_t, j_t) \leq \varepsilon$, then $i_t \overset{G^\star}{\sim} j_t$. The new cluster is $\widehat{G}_t = S \cap G_{k(i_t)}$. Since the algorithm is correct in the first $t-1$ steps, $G_{k(i_t)} \subset S$. Thus, $\widehat{G}_t = G_{k(i_t)}$.

The PARTITION algorithm is correct in both cases at step t, which completes the induction. □

The quantity τ is the sampling error of correlation differences. Lemma 1 implies that if τ is small enough, the PARTITION algorithm recovers the partition $G^\star$ with a properly chosen ε. To prove Theorem 3, we also need the following lemma, which gives an estimate of τ.

Lemma 2. *Under Assumption 1, there exist numerical constants* $c_1, c_2 > 0$ *such that*

$$\tau \leq 2L^2\left(c_1\sqrt{\frac{\log d}{n}} + c_2\frac{(\log d)^{\frac{2}{\alpha}}}{n}\right),$$

with probability at least $1 - 4/d$.

The proof of Lemma 2 is based on the following result on the concentration of quadratic forms in i.i.d. random variables with α-sub-exponential distribution. Recall that $\boldsymbol{X}^*$ is the $n \times d$ matrix whose row r is $X^{*r} = (X_1^{*r}, \ldots, X_d^{*r})$, the standardized returns in period $r \in [n]$.

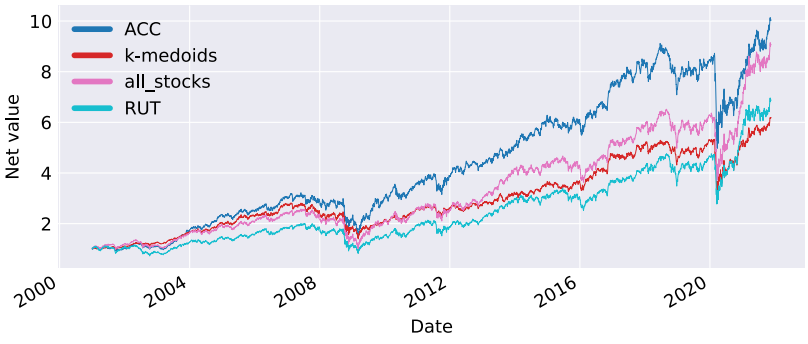
Lemma 3. *Under Assumption 1, there exists* $c > 0$ *such that for any* $t > 0$ *and* $u, v \in \mathbb{R}^d$, *we have*

$$\left|\frac{1}{n}u^\top \boldsymbol{X}^{*\top}\boldsymbol{X}^* v - u^\top \boldsymbol{\rho} v\right| \leq (\ln 2)^{-2/\alpha} L^2 \sqrt{u^\top \boldsymbol{\rho} u}\sqrt{v^\top \boldsymbol{\rho} v}\left(c\sqrt{\frac{t}{n}} + c^{\frac{4}{\alpha}}\frac{t^{\frac{2}{\alpha}}}{n}\right),$$

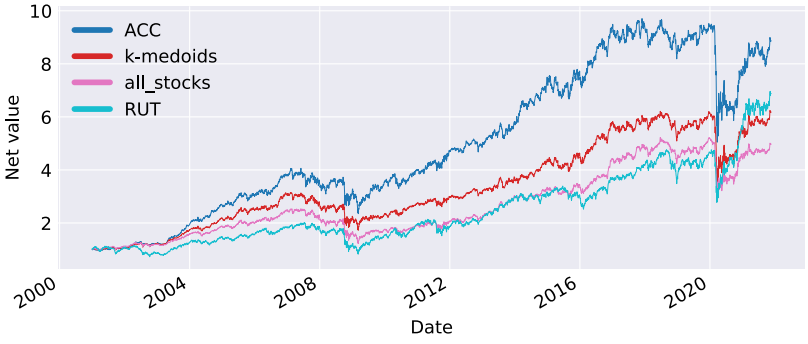
with probability at least $1 - 4e^{-t}$.

The proof of Lemma 3 relies on the following concentration inequality of quadratic forms in i.i.d. random variables with α-sub-exponential distribution.

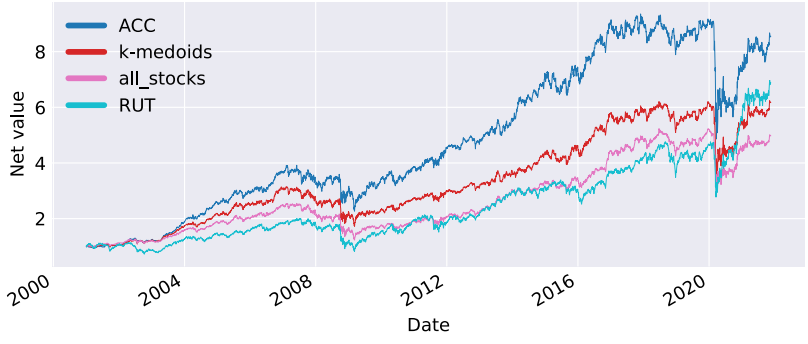
Lemma 4. *(Götze et al., 2021; Sambale, 2020) For* $\alpha \in (0, 2]$, *let* $Y = (Y_1, \ldots, Y_n)$ *be centered and independent random variables such that* $\|Y_r\|_{\psi_\alpha} \leq L$ *for each* $r \in [n]$. *Let* $\boldsymbol{A} = (a_{ij})_{i,j\in[n]\times[n]}$ *be a symmetric matrix.*



(a) Risk parity portfolios



(b) Minimum variance portfolios



(c) Single-period mean-variance portfolios with 10% target return

Fig. 11. Daily value comparison among the portfolios. ACC chooses between 15 and 25 stocks, and k-medoids chooses 20 stocks. Rebalanced annually.

Then there exists a constant $C = C(\alpha)$ such that for any $t > 0$,

$$\mathbb{P}\left(|Y^\top \boldsymbol{A} Y - \mathbb{E}(Y^\top \boldsymbol{A} Y)| > t\right) \le 2\exp\left(-C\min\left(\frac{t^2}{L^4\|\boldsymbol{A}\|_2^2}, \left(\frac{t}{L^2\|\boldsymbol{A}\|_{op}}\right)^{\frac{\alpha}{2}}\right)\right), \tag{15}$$

where $\|\boldsymbol{A}\|_2 := \sqrt{\sum_{i,j=1}^n a_{ij}^2}$ is the 2-norm, and $\|\boldsymbol{A}\|_{op} := \sup\{|\boldsymbol{A}x| : |x| = 1\}$ is the operator norm.

See also Adamczak (2015), Jeong et al. (2020), Rudelson and Vershynin (2013), Vu and Wang (2015) for related results on concentration inequalities of quadratic forms in i.i.d. random variables. Now we proceed to prove Lemma 3.

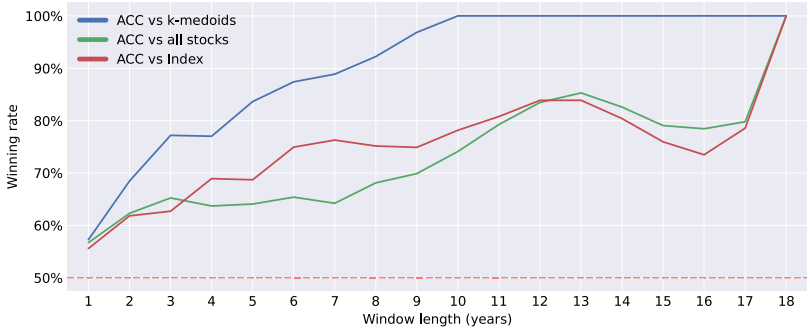
Proof of Lemma 3. First, we express the inequality (15) in a slightly different way. By replacing t with $s := cL^2(\|\boldsymbol{A}\|_2\sqrt{t} + c^{\frac{4}{\alpha}-1}\|\boldsymbol{A}\|_{op}t^{\frac{2}{\alpha}})$ where $c > 0$ is some constant and t is any positive number, we have

$$\begin{aligned}
&\mathbb{P}(|Y^\top \boldsymbol{A} Y - \mathbb{E}(Y^\top \boldsymbol{A} Y)| \ge s) \\
&\le 2\exp\left(-C\min\left(\frac{s^2}{L^4\|\boldsymbol{A}\|_2^2}, \left(\frac{s}{L^2\|\boldsymbol{A}\|_{op}}\right)^{\frac{\alpha}{2}}\right)\right) \\
&\le 2\exp\left(-C\min\left(\frac{(cL^2\|\boldsymbol{A}\|_2\sqrt{t})^2}{L^4\|\boldsymbol{A}\|_2^2}, \left(\frac{cL^2c^{\frac{4}{\alpha}-1}\|\boldsymbol{A}\|_{op}t^{\frac{2}{\alpha}}}{L^2\|\boldsymbol{A}\|_{op}}\right)^{\frac{\alpha}{2}}\right)\right) \\
&= 2\exp(-Cc^2t).
\end{aligned}$$

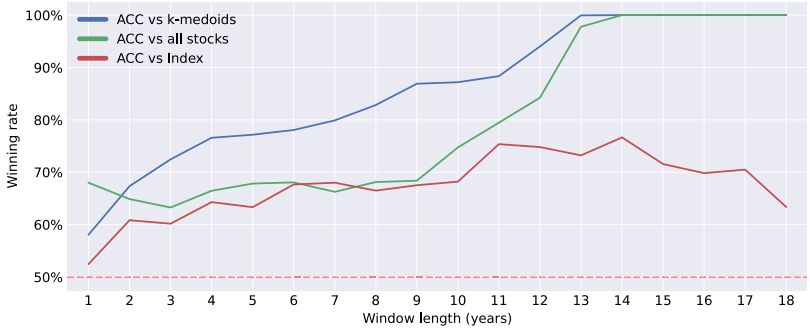
By taking $c = \sqrt{1/C}$, we get, for any $t > 0$,

$$\mathbb{P}\left(\left|Y^\top \boldsymbol{A} Y - \mathbb{E}\left(Y^\top \boldsymbol{A} Y\right)\right| \ge cL^2\left(\|\boldsymbol{A}\|_2\sqrt{t} + c^{\frac{4}{\alpha}-1}\|\boldsymbol{A}\|_{op}t^{\frac{2}{\alpha}}\right)\right) \le 2e^{-t}. \tag{16}$$

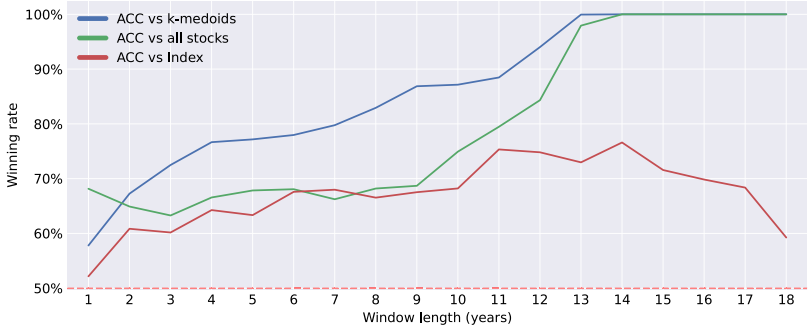
By Assumption 1, for any $\omega \in \mathbb{R}^d$ and $r \in [n]$, we have $\|(X^{*r})^\top\omega\|_{\psi_\alpha} = \|(\boldsymbol{\rho}^{-1/2}X^{*r})^\top(\boldsymbol{\rho}^{1/2}\omega)\|_{\psi_\alpha} \le \|\boldsymbol{\rho}^{-1/2}X^{*r}\|_{\psi_\alpha}\|\boldsymbol{\rho}^{1/2}\omega\|_{\psi_\alpha} \le L(\ln 2)^{-1/\alpha}\sqrt{\omega^\top\boldsymbol{\rho}\omega}$.



(a) Risk parity



(b) Minimum variance



(c) Mean-variance

Fig. 12. Percentage of sub-periods in which ACC has a higher return than benchmarks under different allocation strategies.

By applying (16) to $Y = \boldsymbol{X}^{*\top}\omega$ and $\boldsymbol{A} = \boldsymbol{I}_n$ with $\omega = \lambda u + \lambda^{-1}v$ and $\omega = \lambda u - \lambda^{-1}v$ for some $\lambda > 0$, respectively, we get

$$\left||\lambda\boldsymbol{X}^*u \pm \lambda^{-1}\boldsymbol{X}^*v|^2 - \mathbb{E}\left[|\lambda\boldsymbol{X}^*u \pm \lambda^{-1}\boldsymbol{X}^*v|^2\right]\right|$$
$$\leq (\ln 2)^{-2/\alpha} c L^2 (\lambda u \pm \lambda^{-1}v)^\top \boldsymbol{\rho} (\lambda u \pm \lambda^{-1}v)(\sqrt{nt} + c^{\frac{4}{\alpha}-1} t^{\frac{2}{\alpha}}),$$

with probability at least $1 - 4e^{-t}$. Notice that $u^\top \boldsymbol{X}^{*\top}\boldsymbol{X}^* v = \frac{1}{4}(|\lambda\boldsymbol{X}^*u + \lambda^{-1}\boldsymbol{X}^*v|^2 - |\lambda\boldsymbol{X}^*u - \lambda^{-1}\boldsymbol{X}^*v|^2)$. As a consequence,

$$\begin{aligned}\left|\frac{1}{n}u^\top \boldsymbol{X}^{*\top}\boldsymbol{X}^* v - u^\top \boldsymbol{\rho} v\right| &\leq \frac{1}{4n}\Big(\Big||\lambda\boldsymbol{X}^*u + \lambda^{-1}\boldsymbol{X}^*v|^2 - \mathbb{E}\left[|\lambda\boldsymbol{X}^*u + \lambda^{-1}\boldsymbol{X}^*v|^2\right]\Big| \\ &\quad + \Big||\lambda\boldsymbol{X}^*u - \lambda^{-1}\boldsymbol{X}^*v|^2 - \mathbb{E}\left[|\lambda\boldsymbol{X}^*u - \lambda^{-1}\boldsymbol{X}^*v|^2\right]\Big|\Big) \\ &\leq \frac{1}{4n}(\ln 2)^{-2/\alpha} c L^2 (\sqrt{nt} + c^{\frac{4}{\alpha}-1} t^{\frac{2}{\alpha}})\left[2\lambda^2 u^\top \boldsymbol{\rho} u + 2\lambda^{-2} v^\top \boldsymbol{\rho} v\right].\end{aligned}$$

By taking $\lambda = (v^\top \boldsymbol{\rho} v / u^\top \boldsymbol{\rho} u)^{1/4}$, we get the desired result. □

Now we prove Lemma 2.

Proof of Lemma 2. For $i, j, l \in [d]$, let $u = e_i - e_j$ and $v = e_l$, where $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ with the ith coordinate one and all others zeros. It is easy to see that

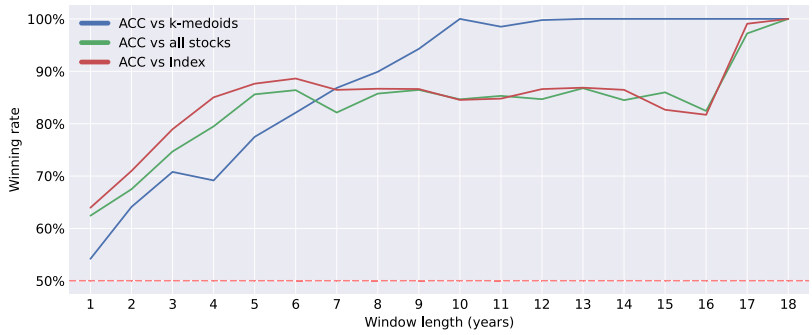
$$\widehat{\rho}_{il} - \widehat{\rho}_{jl} = \frac{1}{n}u^\top \boldsymbol{X}^{*\top}\boldsymbol{X}^* v \quad \text{and} \quad \rho_{il} - \rho_{jl} = u^\top \boldsymbol{\rho} v.$$

By Lemma 3, we have for any $t > 0$,

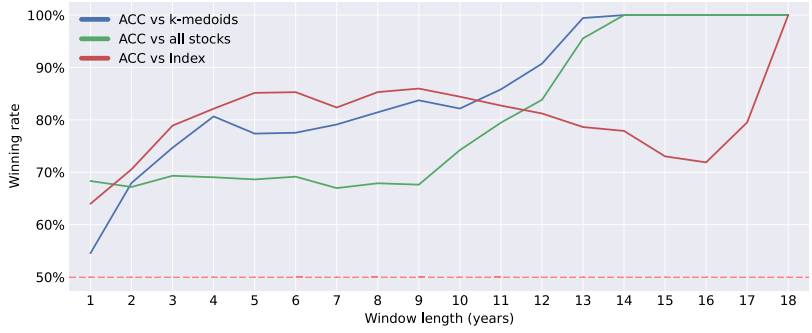
$$\begin{aligned}|(\widehat{\rho}_{il} - \widehat{\rho}_{jl}) - (\rho_{il} - \rho_{jl})| &\leq (\ln 2)^{-2/\alpha} L^2 \sqrt{2 - 2\rho_{ij}}\left(c\sqrt{\frac{t}{n}} + c^{\frac{4}{\alpha}} \frac{t^{\frac{2}{\alpha}}}{n}\right) \\ &\leq 2(\ln 2)^{-2/\alpha} L^2 \left(c\sqrt{\frac{t}{n}} + c^{\frac{4}{\alpha}} \frac{t^{\frac{2}{\alpha}}}{n}\right),\end{aligned}$$

with probability at least $1 - 4e^{-t}$. Notice that the above inequality holds for any $1 \leq i < j \leq d$ and $l \neq i, j$. Taking $t = \log(d)$, we have

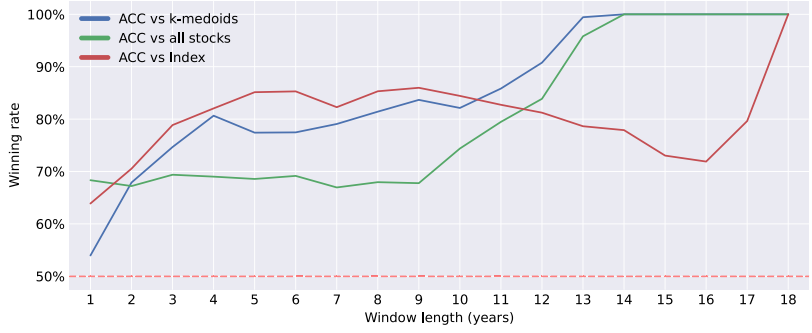
$$|(\widehat{\rho}_{il} - \widehat{\rho}_{jl}) - (\rho_{il} - \rho_{jl})| \leq 2(\ln 2)^{-2/\alpha} L^2 \left(c\sqrt{\frac{\log d}{n}} + c^{\frac{4}{\alpha}} \frac{(\log d)^{\frac{2}{\alpha}}}{n}\right),$$



(a) Risk parity



(b) Minimum variance



(c) Mean-variance

Fig. 13. Percentage of sub-periods in which ACC has a higher Sharpe ratio than benchmarks under different allocation strategies.

for any $1 \le i < j \le d$ and $l \ne i, j$, with probability at least $1 - 4/d$. Let $c_1 = (\ln 2)^{-2/\alpha} c$ and $c_2 = (\ln 2)^{-2/\alpha} c^{4/\alpha}$, we have

$$\tau = \max_{i,j,l \in [d]} |(\widehat{\rho}_{il} - \widehat{\rho}_{jl}) - (\rho_{il} - \rho_{jl})| \le 2L^2 \left(c_1 \sqrt{\frac{\log d}{n}} + c_2 \frac{(\log d)^{\frac{2}{\alpha}}}{n} \right)$$

with probability at least $1 - 4/d$. □

Finally, Theorem 3 follows easily from Lemma 1 and Lemma 2.

A.4. Proof of Theorem 4

The ACC algorithm is decomposed into two stages.

The first is the preparation stage, where the sample correlation matrix and the $\widehat{\text{CORD}}$ matrix are computed, and the tail parameters are estimated. The standardization step takes $\mathcal{O}(nd)$. The complexity of computing $\widehat{\boldsymbol{\rho}}$ is $\mathcal{O}(nd^2)$. For any $i, j \in [d]$, the complexity of computing $\widehat{\text{CORD}}(i, j)$ is $\mathcal{O}(d)$. So computing the entire $\widehat{\text{CORD}}$ matrix at most $d^2 \cdot \mathcal{O}(d) = \mathcal{O}(d^3)$ operations. Calculating $\widehat{\boldsymbol{\rho}}^{-1/2}$ can be done through eigenvalue decomposition, which has complexity $\mathcal{O}(d^3)$. Sorting Y_i takes $\mathcal{O}(n \log(n))$, and each linear regression takes $\mathcal{O}(n)$. Therefore, the overall complexity of the preparation stage is $\mathcal{O}(d^2(n + d))$.

The second stage is the grid search stage, where an appropriate ε is chosen based on the results of the PARTITION procedure using different values for ε. In the PARTITION procedure, the while loop has at most d iterations. In each iteration, finding $\text{argmin}_{i,j \in S, i \ne j} \widehat{\text{CORD}}(i, j)$ would take $\mathcal{O}(d^3)$, but it can be simplified by keeping a sorted list of all values in $\widehat{\text{CORD}}$, since the same $\widehat{\text{CORD}}$ is passed to PARTITION every time. With this sorted list, finding $\text{argmin}_{i,j \in S, i \ne j} \widehat{\text{CORD}}(i, j)$ is at most $\mathcal{O}(d)$. Finding the set $\left\{k \in S : \min\left(\boldsymbol{D}(i_l, k), \boldsymbol{D}(j_l, k)\right) \le \varepsilon\right\}$ is at most $\mathcal{O}(d)$, and all other steps are constant. Overall, each PARTITION call takes $\mathcal{O}(d^2)$. Sorting all entrees in $\widehat{\text{CORD}}$ in advance takes $\mathcal{O}(d^2 \log(d^2)) = \mathcal{O}(d^3)$. Because the number of grids n_g in the grid search is a constant and does not grow with d or n, the grid search stage has complexity $\mathcal{O}(d^3)$.

Therefore, the overall complexity of Algorithm 1 is $\mathcal{O}(nd^2 + d^3)$.

Appendix B. Experimental details

In this section, we recall some algorithms used in our empirical study.

Table 9
Performance metrics of the risk parity portfolios. ACC chooses between 15 and 25 stocks, and k-medoids chooses 20 stocks. Rebalanced annually.

	ACC	k-medoids	All stocks	Russel 2000 Total Return Index
Ending VAMI	9823.57	5965.45	8799.78	6720.99
Max Drawdown	54.68%	50.31%	60.68%	58.89%
Peak-To-Valley	2007-06-04–2009-03-09	2007-02-07–2009-03-09	2007-06-04–2009-03-09	2007-07-13–2009-03-09
Recovery	252 Days	964 Days	449 Days	488 Days
Sharpe Ratio	0.55	0.47	0.48	0.39
Sortino Ratio	0.88	0.74	0.77	0.63
Calmar Ratio	0.21	0.18	0.18	0.16
Ann. Volatility	21.17%	19.07%	22.89%	24.32%
Ann. Downside Volatility	13.13%	12.14%	14.29%	15.24%
Correlation	0.91	0.9	0.99	1.0
Beta	0.79	0.71	0.93	1.0
Ann. Return	11.61%	8.97%	11.02%	9.59%
Ann. Turnover Ratio	70.25%	70.09%	27.16%	–
Positive Periods	2778 (53.01%)	2756 (52.59%)	2814 (53.69%)	2800 (53.42%)
Negative Periods	2463 (46.99%)	2485 (47.41%)	2427 (46.31%)	2441 (46.58%)

Table 10
Performance metrics of the minimum variance portfolios. ACC chooses between 15 and 25 stocks, and k-medoids chooses 20 stocks. Rebalanced annually.

	ACC	k-medoids	All stocks	Russel 2000 Total Return Index
Ending VAMI	8679.42	5978.93	4826.0	6720.99
Max Drawdown	47.99%	45.25%	51.71%	58.89%
Peak-To-Valley	2017-11-29–2020-03-23	2007-02-07–2009-03-09	2007-06-04–2009-03-09	2007-07-13–2009-03-09
Recovery	–	867 Days	1085 Days	488 Days
Sharpe Ratio	0.56	0.48	0.42	0.39
Sortino Ratio	0.88	0.74	0.66	0.63
Calmar Ratio	0.23	0.2	0.15	0.16
Ann. Volatility	19.65%	18.86%	18.69%	24.32%
Ann. Downside Volatility	12.43%	12.1%	11.93%	15.24%
Correlation	0.81	0.83	0.88	1.0
Beta	0.65	0.64	0.68	1.0
Ann. Return	10.95%	8.98%	7.86%	9.59%
Ann. Turnover Ratio	62.09%	64.81%	63.63%	–
Positive Periods	2774 (52.93%)	2757 (52.60%)	2777 (52.99%)	2800 (53.42%)
Negative Periods	2467 (47.07%)	2484 (47.40%)	2464 (47.01%)	2441 (46.58%)

Table 11
Performance metrics of the single period mean–variance portfolios. ACC chooses between 15 and 25 stocks, and k-medoids chooses 20 stocks. Rebalanced annually.

	ACC	k-medoids	All stocks	Russel 2000 Total Return Index
Ending VAMI	8369.0	5973.96	4830.79	6720.99
Max Drawdown	48.0%	45.25%	51.72%	58.89%
Peak-To-Valley	2017-11-29–2020-03-23	2007-02-07–2009-03-09	2007-06-04–2009-03-09	2007-07-13–2009-03-09
Recovery	–	867 Days	1085 Days	488 Days
Sharpe Ratio	0.55	0.48	0.42	0.39
Sortino Ratio	0.87	0.74	0.66	0.63
Calmar Ratio	0.22	0.2	0.15	0.16
Ann. Volatility	19.64%	18.86%	18.67%	24.32%
Ann. Downside Volatility	12.43%	12.1%	11.92%	15.24%
Correlation	0.81	0.83	0.88	1.0
Beta	0.65	0.64	0.68	1.0
Ann. Return	10.76%	8.97%	7.87%	9.59%
Ann. Turnover Ratio	62.31%	64.77%	63.54%	–
Positive Periods	2781 (53.06%)	2756 (52.59%)	2777 (52.99%)	2800 (53.42%)
Negative Periods	2460 (46.94%)	2485 (47.41%)	2464 (47.01%)	2441 (46.58%)

B.1. Single-linkage clustering

Single-linkage is an agglomerative hierarchical clustering method and follows the following general procedure on a given distance matrix:

1. Begin with d clusters, each consisting of exactly one entity. Label the clusters with the numbers $1, \dots, d$.
2. Search the distance matrix for the closest pair of clusters. Let the chosen clusters be labeled A and B, and their distance be $d_{A,B}$. This distance $d_{A,B}$ is usually based on distances between members of clusters A and B.
3. Merge clusters A and B, thus reducing the total number of clusters by 1. Relabel the merged cluster as A, and update the distance matrix to reflect the revised distance between cluster A and all other existing clusters. Delete the row and column in the distance matrix pertaining to cluster B.
4. Repeat Steps 2 and 3 until the desired number of clusters are obtained.

Different agglomerative hierarchical clustering algorithms have been proposed with different definitions of the distance measure $d_{A,B}$. In the single-linkage, the distance between two clusters is defined as the distance between the two closest members of clusters i and j:

$$d_{A,B} := \min_{a\in A, b\in B} D(a,b),$$

where $D(a,b)$ is a given distance between entities a and b.[12] In Mantegna (1999), Steps 2 and 3 in the single-linkage clustering is repeated until only one cluster remains. By recording $\operatorname{argmin}_{a\in A,b\in B} D(a,b)$ in each merge, a maximum spanning tree is obtained, together with a hierarchical organization among all entities.

B.2. k-medoids clustering

The k-medoids clustering method (Kaufman & Rousseeuw, 1990) aims to find clusters of similar entities by first identifying a set of k representative entities. Then, k clusters are constructed by assigning each entity to the nearest representative object. Such representative objects are called *medoids* of the clusters, hence the name k-medoids. The k-medoids method can be formulated as the following optimization problem:

$$\min_{M}\left(\sum_{i\in[d], i\notin M}\min_{m\in M} D(i,m)\right)$$

subject to $M \subset [d]$

$$|M| = k.$$

Here, M denotes the set of medoids, which must be a subset of all entities $[d]$ and must have cardinality k. $D(a,b)$ is a given distance between entities a and b. For any given set of medoids M, we assign each non-medoid entity to its closest medoid m, and the optimization finds such a set of medoids that minimizes the shortest total distance between non-medoid entities and their corresponding medoids.

The k-medoid is implemented as an iterative algorithm that gradually improves the quality of M. First, a set of initial medoids is chosen. Different initialization methods can be applied here. Kaufman and Rousseeuw (1990) propose their own initialization, where k medoids are selected in sequence such that the first medoid is the most centered entity, and each subsequent medoid decreases the objective function as much as possible. In practice, random initialization is often used for simplicity. In our implementation, the first medoid is randomly selected, and then each subsequent medoid is the entity with the largest distance from its closest existing medoid.[13]

After initialization, the algorithm improves the medoids by considering all possible swaps, i.e., replacing a medoid h with a non-medoid entity i, and carrying out the swap that decreases the objective function as much as possible. The algorithm stops when no swap can decrease the objective function anymore.

In our experiments, the distance measure between stocks a and b used in both the hierarchical clustering method and the k-medoids method is $D(a,b) := \sqrt{2(1-\rho_{ab})}$, the same as in Mantegna (1999), and the desired number of clusters k is set to be 20 for those two methods.

B.3. Risk parity

The concept of risk parity was pioneered by Bridgewater in its All Weather strategy launched in 1996.[14] In our experiments, we apply the version of risk parity that equalizes weighted marginal risk contribution, also adopted by Maillard et al. (2010), of every asset in the portfolio. To be more precise, define the volatility of a portfolio

$$\sigma(w) = \sqrt{w^T \boldsymbol{\Sigma} w}, \tag{17}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix, and w is the vector of allocation weights of d assets of the portfolio. Hence, the risk contribution of asset i is

$$\sigma_i(w) = w_i \frac{\partial \sigma(w)}{\partial w_i} = \frac{w_i(\boldsymbol{\Sigma} w)_i}{\sigma(w)}. \tag{18}$$

Observe that $\sigma(w) = \sum_{i=1}^{d} \sigma_i(w)$. We now construct a portfolio in such a way that the risk contribution of each asset is equal, namely

$$\sigma_i(w) = \frac{\sigma(w)}{d} \iff w_i = \frac{\sigma(w)^2}{d\cdot(\boldsymbol{\Sigma} w)_i}. \tag{19}$$

It is easy to see that the problem (19) is equivalent to the non-linear optimization problem

$$\begin{aligned} &\min_{w} \sum_{i=1}^{d}\left(w_i - \frac{\sigma(w)^2}{d\cdot(\boldsymbol{\Sigma} w)_i}\right)^2 \\ &\text{subject to } \sum_{i=1}^{d} w_i = 1,\ w_i > 0. \end{aligned} \tag{20}$$

B.4. Markowitz's mean–variance strategy and minimum variance strategy

Markowitz's original mean–variance strategy without short-selling:

$$\begin{aligned} \min\quad & w^{\top}\boldsymbol{\Sigma} w \\ \text{s.t.}\quad & w^{\top}\mu \geq \alpha \\ & w^{\top}1 = 1,\ w \geq 0, \end{aligned} \tag{21}$$

where $\mu \in \mathbb{R}^d$ contains the mean returns of the stocks, which is estimated using the average of the daily returns in the backward-looking window, and α is the target return, which we set to 10%.

The minimum variance strategy is similar to Markowitz's mean–variance optimization but without the expected return constraint:

$$\begin{aligned} \min\quad & w^{\top}\boldsymbol{\Sigma} w \\ \text{s.t.}\quad & w^{\top}1 = 1,\ w \geq 0. \end{aligned} \tag{22}$$

[12] Similar agglomerative hierarchical clustering algorithms include complete-linkage: $d_{A,B} := \max_{a\in A,b\in B} D(a,b)$, and average-linkage: $d_{A,B} := \frac{1}{|A||B|}\sum_{a\in A,b\in B} D(a,b)$. For a more detailed discussion we refer to Anderberg (1973).

[13] We use the python package `pyclustering` for this initialization and the subsequent k-medoid clustering.

[14] "The All Weather Story", Bridgewater, Accessed February 25, 2021. https://www.bridgewater.com//_document?id=00000171-8623-d7de-affd-feaf4ee20000.

Appendix C. Backtesting on Russell 2000

Here, we report our backtesting results on the constituents of the Russell 2000 Index. On the first trading day of each year, we collect the point-in-time constituents, select stocks with at least 7 years of history and no more than 5% of missing data in the past 1750 trading days, and then compute clustering and asset allocations using data in the same 1750-day window. The increase in the length of the backward-looking window compared to the S&P 500 experiments is to accommodate for the larger number of eligible stocks, which ranges from 951 to 1408 in the years between 2001 and 2021. We still let ACC recover between 15 and 25 clusters and let k-medoids recover 20 stocks, thus keeping the size of the final portfolios small. We create portfolios consisting of the lowest-variance stock from each cluster, based on the results of ACC and k-medoids algorithms. For comparison, we also create a portfolio consisting of all eligible stocks and include the Russell 2000 Total Return Index (shortened as RUT) as benchmarks.

Fig. 11 compares the daily performance of these portfolios, and Tables 9–11 display some detailed metrics. Our ACC algorithm consistently produces higher Sharpe ratios than the Russell 2000 Index, the portfolio consisting of all eligible stocks, as well as the portfolio based on the k-medoids clustering results. In more comprehensive comparisons shown in Figs. 12 and 13, ACC has a consistent outperformance against k-medoids, all-stock portfolio, and the benchmark index in different sub-periods, in terms of both the total return and the Sharpe ratio.

References

Abbe, E. (2017). Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, *18*, Paper No. 177, 86.

Adamczak, R. (2015). A note on the hanson-wright inequality for random vectors with dependencies. *Electronic Communications in Probability*, *20*, no. 72, 13.

Anderberg, M. R. (1973). Chapter 6 - hierarchical clustering methods. In *Probability and mathematical statistics, No. 19*, *Cluster analysis for applications* (pp. 131–155). Academic Press, [ISSN: 00795607].

Brodie, J., Daubechies, I., De Mol, C., Giannone, D., & Loris, I. (2009). Sparse and stable markowitz portfolios. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(30), 12267–12272.

Bunea, F., Giraud, C., & Luo, X. (2016). Minimax optimal variable clustering in G-models via cord. arXiv:1508.01939v2.

Bunea, F., Giraud, C., Luo, X., Royer, M., & Verzelen, N. (2020). Model assisted variable clustering: Minimax-optimal recovery and algorithms. *The Annals of Statistics*, *48*(1), 111–137.

Chaudhury, M. (2014). How did the financial crisis affect daily stock returns? *Journal of Investigative*, *23*(3), 65–84.

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, *1*(2), 223–236.

Das, N. (2003). Hedge fund classification using K-means clustering method. In *9th International conference on computing in economics and finance* (pp. 1–27). URL https://depts.washington.edu/sce2003/Papers/284.pdf.

De Prado, M. L. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, *42*(4), 59–69.

DeMiguel, V., Garlappi, L., Nogales, F. J., & Uppal, R. (2009). A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, [ISSN: 00251909] *55*(5), 798–812. http://dx.doi.org/10.1287/mnsc.1080.0986.

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *Review of Financial Studies*, *22*(5), 1915–1953.

Faaland, B. (1974). An integer programming algorithm for portfolio selection. *Management Science*, [ISSN: 0025-1909] *20*(10), 1376–1384. http://dx.doi.org/10.1287/mnsc.20.10.1376, URL https://pubsonline.informs.org/doi/abs/10.1287/mnsc.20.10.1376.

Gao, J., & Li, D. (2013). Optimal cardinality constrained portfolio selection. *Operations Research*, [ISSN: 0030364X] *61*(3), 745–761. http://dx.doi.org/10.1287/opre.2013.1170, URL https://pubsonline.informs.org/doi/abs/10.1287/opre.2013.1170.

Gardes, L., & Girard, S. (2008). Estimation of the Weibull tail-coefficient with linear combination of upper order statistics. *Journal of Statistical Planning and Inference*, *138*(5), 1416–1427.

Gavrilov, M., Anguelov, D., Indyk, P., & Motwani, R. (2000). Mining the stock market: Which measure is best. In *Proceedings of the 6Th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 487–496).

Götze, F., Sambale, H., & Sinulis, A. (2021). Concentration inequalities for polynomials in α-sub-exponential random variables. *Electronic Journal of Probability*, *26*, http://dx.doi.org/10.1214/21-ejp606, Paper No. 48, 22. 4247973.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Springer series in statistics*, *The elements of statistical learning* (2nd ed.). (p. xxii+745). Springer, New York, Data mining, inference, and prediction.

He, H., Chen, J., Jin, H., & Chen, S. H. (2007). Trading strategies based on K-means clustering and regression models. In *Computational intelligence in economics and finance: Volume II* (pp. 123–134). Springer Berlin.

Ho, M., Sun, Z., & Xin, J. (2015). Weighted elastic net penalized mean-variance portfolio design and computation. *SIAM Journal on Financial Mathematics*, *6*(1), 1220–1244.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Jeong, H., Li, X., Plan, Y., & Yılmaz, O. (2020). Sub-Gaussian matrices on sets: Optimal tail dependence and applications. arXiv:2001.10631.

Kaufman, L., & Rousseeuw, P. J. (1990). Partitioning around medoids (program PAM). In *Finding groups in data* (pp. 68–125). John Wiley & Sons, Ltd.

Korzeniewski, J. (2018). Efficient stock portfolio construction by means of clustering. *Acta Universitatis Lodziensis. Folia Oeconomica*, *1*(333), 85–92.

Krasnoselsky, M. A., & Rutitsky, Y. B. (1961). *Convex functions and orlicz spaces* (p. xi+249). P. Noordhoff Ltd., Groningen.

León, D., Aragón, A., Sandoval, J., Hernández, G., Arévalo, A., & Niño, J. (2017). Clustering algorithms for risk-adjusted portfolio construction. In *Procedia computer science. Vol. 108* (pp. 1334–1343). Elsevier B.V..

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transaction on Information Theory*, *28*(2), 129–137.

Maillard, S., Roncalli, T., & Teïletche, J. (2010). The properties of equally weighted risk contribution portfolios. *Journal of Portfolio Management*, *36*(4), 60–70.

Mantegna, R. N. (1999). Hierarchical structure in financial markets. *The European Physical Journal B*, *11*(1), 193–197, arXiv:9802256.

Marathe, A., & Shawky, H. A. (1999). Categorizing mutual funds using clusters. *Advances In Quantitative Analysis Of Finance And Accounting*, *7*(1), 199–204.

Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77–91.

Markowitz, H. M. (1959). *Cowles foundation for research, monograph 16*, *Portfolio selection: efficient diversification of investments* (p. x+344). New York: John Wiley & Sons, Inc.

Marvin, K. (2015). *Creating diversified portfolios using cluster analysis. July*. Princeton University, URL https://www.cs.princeton.edu/sites/default/files/uploads/karina_marvin.pdf.

McNamee, J. L., Paradise, T., & Bruno, M. A. (2019). *Financial planning perspectives getting back on track: a guide to smart rebalancing*: *Technical report vanguard research*, URL https://personal.vanguard.com/pdf/ISGGBOT.pdf.

Merton, R. C. (1980). On estimating the expected return on the market: an exploratory investigation. *Topics In Catalysis*, *8*(4), 323–361.

Musmeci, N., Aste, T., & Di Matteo, T. (2015). Relation between financial market structure and the real economy: Comparison between clustering methods. *PLoS One*, *10*(3), 1–29, arXiv:arXiv:1406.0496v1.

Nakagawa, K., Imamura, M., & Yoshida, K. (2019). Stock price prediction using k-medoids clustering with indexing dynamic time warping. *Electronics And Communications In Japan*, *102*(2), 3–8.

Nanda, S. R., Mahanty, B., & Tiwari, M. K. (2010). Clustering indian stock market data for portfolio management. *Expert Systems With Applications*, *37*(12), 8793–8798.

Pozzi, F., Di Matteo, T., & Aste, T. (2013). Spread of risk across financial markets: Better to invest in the peripheries. *Scientific Reports*, *3*(1), 1–7.

Puerto, J., Rodríguez-Madrena, M., & Scozzari, A. (2020). Clustering and portfolio selection problems: A unified framework. *Computers And Operations Research*, [ISSN: 03050548] *117*, Article 104891. http://dx.doi.org/10.1016/j.cor.2020.104891.

Raffinot, T. (2017). Hierarchical clustering-based asset allocation. *The Journal of Portfolio Management*, *44*(2), 89–99.

Reilly, F. K., & Brown, K. C. (2012). An introduction to asset pricing models. In *Investment analysis & portfolio management* (10th ed.). South-Western Cengage Learning.

Ren, Z. (2005). Portfolio construction using clustering methods. In *Animal genetics*. Worcester Polytechnic Institute, URL https://digital.wpi.edu/concern/etds/cr56n106c?locale=en.

Rosén, F. (2006). Correlation based clustering of the Stockholm Stock Exchange. In *Business administration, stockholm university*. Stockholm University, URL http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A196577&dswid=-2241.

Rudelson, M., & Vershynin, R. (2013). Hanson-wright inequality and sub-Gaussian concentration. *Electronic Communications in Probability*, *18*, no. 82, 9.

Sambale, H. (2020). Some notes on concentration for α-subexponential random variables. arXiv:2002.10761.

Tumminello, M., Aste, T., Di Matteo, T., & Mantegna, R. N. (2005). A tool for filtering information in complex systems. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(30), 10421–10426.

Vladimirova, M., Girard, S., Nguyen, H., & Arbel, J. (2020). Sub-Weibull distributions: Generalizing sub-Gaussian and sub-exponential properties to heavier-tailed distributions. *Stat*, *9*, e318, 8.

Vu, V., & Wang, K. (2015). Random weighted projections, random quadratic forms and random eigenvectors. *Random Structures & Algorithms*, *47*(4), 792–821.

Zaremba, A., & Shemer, J. (2017). Is risk always rewarded? Low-volatility anomalies. In *Country asset allocation: quantitative country selection strategies in global factor investing* (pp. 81–104). New York: Palgrave Macmillan US.

Zhan, H. C. J., Rea, W., & Rea, A. (2015). An application of correlation clustering to portfolio diversification. arXiv:1511.07945.

Zilbering, Y., Jaconetti, C. M., & Kinniry, F. M. (2015). *Best practices for portfolio rebalancing*: *Technical Report Vanguard Research*, URL https://www.vanguardfrance.fr/documents/best-practices-for-portfolio-rebalancing-tlrv.pdf.