Growing semantic vines for robust asset allocation

Frank Z. Xing a, Erik Cambria a,*, Roy E. Welsch b

a School of Computer Science & Engineering, Nanyang Technological University, Singapore
b MIT Sloan School of Management, Massachusetts Institute of Technology, USA

ARTICLE INFO

Article history:
Received 8 May 2018
Received in revised form 27 November 2018
Accepted 30 November 2018
Available online 12 December 2018

Keywords:
Vine
Dependence modeling
Semantic analysis
Asset allocation
Robust estimation
Financial text mining

ABSTRACT

The vine structure has been widely studied as a graphical representation for high-dimensional dependence modeling, depicting complicated probability density functions, and robust correlation estimation. However, specification of the best vine structure is challenging as the number of candidate vine structures grows combinatorially when the number of elements increases. In this article, we propose to leverage semantic prior knowledge of assets extracted from their descriptive documents to find a suitable vine structure for financial portfolio optimization. A vine growing algorithm is provided and the robust covariance matrix estimation process is performed on this vine structure. Our construction of a semantic vine improves the state-of-the-art arbitrary vine-growing method in the context of robust correlation estimation and multi-period asset allocation. The effectiveness of our methods on a large scale is also demonstrated by experiments.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Estimating asset expected returns and correlations is of great importance under the framework of modern portfolio theory, because the expected returns reflect the profitability of the portfolio, and portfolio risk depends on the correlations between pairs of assets. In multivariate analysis, robust estimation of the correlation matrix is non-trivial, especially in high-dimensional cases. Many robust estimators have been proposed for pairwise correlation estimation, however, one cannot guarantee the positive definiteness of a correlation matrix by simply using robust pairwise correlation estimates as matrix elements. When applying this ill-prepared correlation matrix to modern portfolio theory, the risk measurement may become negative. However, this should not happen because the covariance matrix is positive definite by definition. In the situation of negative risk, the optimized portfolio weights will be a corner solution of the feasible domain. Without the constraint of no short selling, the unstable estimation of asset correlation matrix will cause extreme weights and frequent rebalancing of the portfolio. This is why instead of price prediction [1], practitioners are more interested in the robustness of the asset management method [2].

Previous studies have explored many techniques to robustly estimate both expected returns and the covariance matrix, such as minimum covariance determinant (MCD), minimum volume ellipsoid (MVE), M-estimators, shrinkage estimators, iterated bivariate Winsorization (IBW), trimming, and quantile statistics [3–7]. Some of them argue that the expected returns are more influential and harder to estimate because of the existence of exogenous variables [6,8]; others argue that estimating the covariance matrix is more challenging because it involves quadratic times of parameter from the dependence structure of the assets [9].

In this article, we are more interested in a robust estimation of the covariance matrix. Because with the constraints on a vine structure, we can easily guarantee the positive definiteness of our covariance matrix estimation, hence the optimized portfolio weights stay away from corner solutions and unbalanced cases. Under this construction, the critical problem is on how to form a vine from scratch, or how to determine a suitable vine structure from all the permuted possibilities.

Recently, increasing attention has been paid to applying natural language processing (NLP) techniques, such as text mining from the Web [10,11], topic modeling [12], and sentiment analysis from social media [13–16] to financial forecasting and asset management problems [2]. The construction of vine structures requires an understanding of asset properties. The dependence relations of assets are stable when compared to the sentiment-driven price movements and are related to intrinsic factors, such what industry they belong to, their products, and their position in a supply chain. This type of knowledge is formerly absent from the Markowitz model as well as the CAPM [17]. Therefore, we leverage the semantic prior knowledge as a solution to the vine selection problem. The formed vine is later used to estimate covariance matrices robustly.

The main contributions of this paper are:

● an attempt to make use of semantic knowledge of financial assets in the context of vine dependence structure growing,
the first explicitly stated algorithm for robust estimation of correlation matrix on regular semantic vines,¹

- extensive empirical research using stock prices as real-world financial data — benchmarking portfolio on arbitrary vine structures and study of algorithm scalability.

The remainder of this paper is organized as follows: the next sections present challenges of modern portfolio theory, the background of vine structure and document embedding; following, we present a vine growing algorithm and robust asset correlation matrix estimation on the vine structure; later, we demonstrate this process with an example and evaluate the robust portfolio performance; in the end, we discuss our findings and future work.

2. Preliminaries

We recall the prevalently recognized portfolio construction framework by Markowitz [18]. The idea of his mean–variance method is that when the investor will need to allocate a given amount of capital to different assets, he faces an optimization problem of two objectives: (1) he wants to maximize the portfolio expected return, and (2) for a given return level he wants to minimize the risk [18]. If we use the variance of the expected return as a risk metric, we will have an optimization problem as follows:

\[
\begin{align*}
\max_{w_i} & \quad \sum_{i=1}^{N} \mu_i w_i - \frac{\delta}{2} \text{var} \left( \sum_{i=1}^{N} \mu_i w_i \right) \\
\text{s.t.} & \quad \sum_{i=1}^{N} w_i = 1, \quad i = 1, 2, \ldots, N
\end{align*}
\]

where \(\delta\) is an indicator of risk aversion that trades off between the investor’s two objectives, \(w_i\) denotes the weight of the corresponding asset in the portfolio, \(\mu_i\) denotes the expected return on asset \(i\).

We analyze the second item in Eq. (1):

\[
\text{var} \left( \sum_{i=1}^{N} \mu_i w_i \right) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_i \sigma_{ij} w_j = w^\prime \Sigma w
\]

where \(\sigma_{ij}\) is the covariance between the returns on asset \(i\) and asset \(j\), \(w\) is a \(N \times 1\) vector of the portfolio weights, \(\Sigma\) is a \(N \times N\) covariance matrix where the element at the \(i\)th row and \(j\)th column is \(\sigma_{ij}\).

The optimized weights of an efficient portfolio are, therefore, given by the first order condition of Eq. (1):

\[
w^* = (\delta \Sigma)^{-1} \mu
\]

If we set the portfolio expected return to a fixed value \(e\), we can observe one sectional weight combination that satisfies both constraints, and the optimization problem can be restated as:

\[
\begin{align*}
\min_{w_i} & \quad w^\prime \Sigma w \\
\text{s.t.} & \quad 1^\prime w = 1, \quad \mu^\prime w = e.
\end{align*}
\]

Fig. 1 provides an example of 3-dimensional visualization of the optimization problem of Eq. (4). The feasible domain is at the intersection segment of the plane of weight constraint (green) and the plane of expected return constraint (cyan). The heat map from black (minimum) to white (maximum) denotes the range of \(w^\prime \Sigma w\). Therefore, a positive definite \(\Sigma\) provides two good properties: \(w^\prime \Sigma w\) will always be positive, and the minimum is continuous and stable.

However, all these constructions are based on the assumption of normally distributed asset returns. In practice, both \(\Sigma\) and \(\mu\) are difficult to estimate accurately because of the diversified and peculiar distribution of asset returns. Fig. 2 illustrates the distribution of realized daily return ratio of Apple’s stock. Obviously, the typical skewed, heavy-tailed distribution cannot be approximated with a Gaussian distribution. In fact, the standard deviation of returns in Fig. 2 (red) is approximately 4 times that of the Gaussian distribution fitting depicted by the blue curve.

Therefore, the classic Markowitz’s approach of estimating the expected return and the covariance matrix from the historical return time series usually cannot guarantee the positive definiteness of \(\Sigma\), especially in high-dimensional cases. This will cause extreme optimized portfolio weights and frequent rebalancing from multi-period applications. In the following section, we discuss a dependence modeling structure that can be used to induce positive definite estimation of the covariance matrix.

3. Vine dependence structure

Many graphical models have been developed for high-dimensional dependence modeling and joint distribution representation, such as Bayesian belief networks [19] and Markov random fields [20]. Unlike others, a vine is unique for its recursive structure, where the edges of nodes become nodes for the next tree.

Fig. 2. Realized return ratio of Apple Inc’s stock. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 1. Visualization of the portfolio optimization problem. Three dimensions denote holding weights of different financial assets, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
According to Bedford and Cooke [24], the full correlation matrix elements of a partial correlation C-vine or D-vine are computable. Although for C-vines and D-vines an analytical solution can be provided for partial correlation computation between any two nodes, for general regular vines the computation has to be step-by-step on subvines. The following Theorem 2 adapted from Lemma 13 in [24] facilitates this computation. We provide Theorem 2 without a proof.

**Theorem 2.** Let $\Sigma$ be the covariance matrix of $n$ joint normal distributed random variables. Write $\Sigma_A$ for the principal submatrix built from row 1 and row 2 of $\Sigma$, etc. so that

$$\Sigma = \begin{bmatrix} \Sigma_A & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_B \end{bmatrix}.$$ 

Then the conditional distribution of elements 1 and 2 is normal and the covariance matrix has the form:

$$\Sigma_{12|3\ldots n} = \Sigma_A - \Sigma_{AB} \Sigma_B^{-1} \Sigma_{BA}. \tag{5}$$

We further derive Theorem 3 based on Theorem 2.

**Theorem 3.** Consider a subvine of only three nodes 1, 2 and 3, where node 2 is the root. The unconditional correlation of 1 and 3 can be calculated from their correlation conditional on 2 and their partial correlations with 2:

$$\rho_{13} = \rho_{13|2} \sqrt{(1 - \rho_{12}^2)(1 - \rho_{23}^2)} + \rho_{12} \rho_{23}. \tag{6}$$

**Proof.** We write the partial correlation matrix as

$$\rho = \begin{bmatrix} 1 & \rho_{13} & \rho_{12} \\ \rho_{13} & 1 & \rho_{23} \\ \rho_{12} & \rho_{23} & 1 \end{bmatrix}.$$ 

We apply Eq. (5) with $\Sigma_B = [1]$ and

$$\Sigma_A = \begin{bmatrix} 1 & \rho_{13} \\ \rho_{13} & 1 \end{bmatrix}, \quad \Sigma_{AB} = \begin{bmatrix} \rho_{12} \\ \rho_{23} \end{bmatrix}, \quad \Sigma_{13|2} = \begin{bmatrix} \sigma_{12}^2 & \rho_{13}\rho_{12}\sigma_{23} \\ \rho_{13}\rho_{12}\sigma_{23} & \sigma_{23}^2 \end{bmatrix},$$

then we will have:

$$\begin{bmatrix} \sigma_{12}^2 & \rho_{13}\rho_{12}\sigma_{23} \\ \rho_{13}\rho_{12}\sigma_{23} & \sigma_{23}^2 \end{bmatrix} = \begin{bmatrix} 1 & \rho_{13} \\ \rho_{13} & 1 \end{bmatrix}^{-1} - \begin{bmatrix} \rho_{12} \\ \rho_{23} \end{bmatrix} \begin{bmatrix} \rho_{12} & \rho_{23} \end{bmatrix},$$

where we derive the following equations.

$$\sigma_{12}^2 = 1 - \rho_{12}^2 \tag{7}$$

$$\sigma_{23}^2 = 1 - \rho_{23}^2 \tag{8}$$

$$\rho_{13} = \rho_{13|2} \sigma_{12}\sigma_{23} + \rho_{12}\rho_{23} \tag{9}$$

Substituting $\sigma_{12}$ and $\sigma_{23}$ in Eq. (9) with Eqs. (7) and (8), we can get Eq. (6). \qed

### 4. Pairwise semantic linkage

In this section, we describe how to model the correlation between two financial assets via their semantic linkage.

#### 4.1. Distributed document representation

To measure the semantic linkage between two financial assets, we consider a more general problem of representing a piece of text with a vector. Then we transform the problem into a well-studied topic of “the similarity measure for real-valued vectors”.

The bag-of-words (BOW) model has been used for a long time in NLP and information retrieval. The BOW model counts the
frequency of word occurrence and represents a text with these statistics. However, the BOW model has several disadvantages. For instance, natural language has a very large vocabulary, hence the vector representation of text is sparse. The word order information is also lost in the BOW model. If we try to capture this word context information with bag-of-n-grams, the dimension of the vector representation will exponentially explode. Another shortcoming of the BOW model is the semantic gap between different word entries. Similar words, for example strong and powerful, are counted as different dimensions [26]. Whereas in reality, there should be a link between the semantics they carried.

These problems require a denser, fixed-length, continuous real-valued representation of words and texts. With the advance of machine learning, this representation (skip-gram word2vec) was discovered with the neural language model training process [27]. Starting with a sparse word matrix representation that has the length of the vocabulary, a shallow neural network[3] is trained with the learning objective of maximizing the average word occurrence probability given its context window:

$$\max_{U, b} \frac{1}{l} \sum_{k=1}^{l-k} \log p(o_k | o_{k-1}, \ldots, o_{k+k})$$ (10)

where $U$ and $b$ are neural network parameters; $o_1, o_2, \ldots, o_l$ is the word list from training corpus; $k$ is the context window length. The neural network parameters are usually trained using stochastic gradient descent (SGD) where the gradients are obtained via back-propagation of word prediction errors [26,28]. If we define the error as a difference of estimated log probability, the update process is the following until convergence.

$$\Delta(U, b) = -\epsilon \frac{\partial \log p(o_k | o_{k-1}, \ldots, o_{k+k})}{\partial(U, b)}$$ (11)

When the training process is done, the neuron weights give the probability distribution of the word’s context, which can also be interpreted as a fixed-length representation of the word.

Inspired by this word representation idea, [26] proposed a distributed representation for a document that outperforms simply averaging the word vectors. The language model is trained by simultaneous learning of continuous representations for a document and its token. Formally, this means for a given word $o_t$ coming from document $\Omega$, the context will become $(o_{t-k}, \ldots, o_{t+k}, d_t)$ instead, where the additional $d_t$ is the token. Using this modification of Eq. (10), the token somehow is associated with the topic of the document. For this reason, the whole document vector is called distributed memory (DM). Using this embedding, a new document can then be assigned a virtual token from the end of context concatenation that represents the semantic distribution at a document level.

4.2. Similarity

With the document embedding technique, we can build a descriptive document for each asset $a_i$ and compute a vector representation $v(a_i)$ that preserves the semantics. As a result, the semantic linkage well aligns with the vector similarity.

We use the cosine similarity to estimate pairwise semantic linkage for asset $a_i$ and $a_j$:

$$s(a_i, a_j) = \cos(w(a_i), w(a_j))$$
$$= \frac{w(a_i) \cdot w(a_j)}{\|w(a_i)\|_2 \|w(a_j)\|_2}.$$ (12)

The pairwise semantic linkage $s(a_i, a_j)$ is later denoted with a short form $s_ij$. Because the vectors are generated from softmax functions, $s_ij$ is between 0 and 1.

5. Growing the semantic vine

Most previous studies resort to experts to determine the vine structure, or simply assuming a C-vine or D-vine structure, then discuss the properties (such as associated correlations or bivariate copulas) of edges. However, for $n$ elements there exist $4n!12n(n−5)$ regular vines [29], only a few out of which are tailored to the data distribution. Recently, Kurowicka [30] proposed a top-down approach for regular vine growing. For each layer, the edge splitting ensures that the absolute partial correlations corresponding to these nodes are the smallest.4 However, this is only possible when the correlation matrix is fully specified. Since we are unable to robustly estimate the partial correlation matrix, the resulting vine structure might not be robust as well. Therefore, our construction is a bottom-up method of vine growing. We propose to grow a semantic vine by first building edges between assets with a strong semantic linkage. Algorithm 1 describes this process.

Algorithm 1: Growing Semantic Vine Structure.

| Data: asset list $a_i$, pairwise semantic linkage $s_{ij}$, $i, j = 1, 2, \ldots, n$ |
| Result: semantic vine $V_i = \{T_i, N_i, \mathcal{E}(V_i)\}$ |
| 1. list($s_{ij}$) $\leftarrow$ descending sort $s_{ij}(i < j)$; |
| 2. for $k = 1, 2, \ldots, n − 1$ do |
| \quad if $k > 1$ then |
| \quad \quad $N_k = \mathcal{E}_{k-1}$; |
| \quad end |
| \quad repeat |
| \quad \quad if adjacent$(i, j)$ in $N_k$ then |
| \quad \quad \quad $\mathcal{E}_k \leftarrow \max($list$(s_{ij}))$; |
| \quad \quad \quad if $\exists$ loop in $\mathcal{E}_k$ then |
| \quad \quad \quad \quad $\text{discard} \max$(list($s_{ij}$)) from $\mathcal{E}_k$; |
| \quad \quad \quad end |
| \quad \quad delete max(list($s_{ij}$)) from list($s_{ij}$); |
| \quad \quad end |
| \quad until $\exists$ degree$(N_k) = 0$; |
| \quad end |
| 15. return $V_i$; |

Algorithm 1 checks the adjacency of two nodes every time before growing an edge. Consequently, the algorithm ensures that the semantic vine is regular, but it is not necessarily a C-vine or D-vine. For each layer, the termination condition checks the degree of every node. This ensures the semantic vine will be fully connected.

We give the formal definition of a semantic vine as below:

Definition 5. A semantic vine is a regular partial correlation vine where the partial correlation values of edges are estimated from pairwise semantic linkages.

Our semantic vine has the following properties. Each edge in the semantic vine is associated with a value $p_{ij}$ that represents the conditional partial correlation, and is calculated from the semantic linkage $s_{ij}$ between the two assets that the edge is spanning:

$$p_{ij} = (2 \cdot s_{ij} − 1)/(1 + \epsilon).$$ (13)

Because $s_{ij}$ is in the range of $[0, 1]$ and $\epsilon$ is a small positive scaling factor, Eq. (13) maps the semantic linkage to a financial linkage strictly in $[−1, 1]$. In real-life cases, semantic-irrelevant assets are often regarded as safe-haven choices from different industries. Therefore, the asset returns would demonstrate reverse movements.

---

3 Shallow refers to the neural networks that only have one hidden layer of neurons.

4 This is defined as the optimal truncation of vines as a minimum number of edges would have large absolute partial correlations and rest of the edges are assumed insignificant (independent).
5.1. Estimating the robust correlation matrix

Below we discuss the inverse process of vine truncation and selection with a given correlation matrix \([30,31]\), namely estimating the robust correlation matrix from the semantic vine.

It is worth pointing out that neither the pairwise semantic linkage matrix \(S\) nor the partial correlation matrix \(P\) can guarantee positive definiteness. Therefore, it is not a robust process to write \(\rho = P^{-1}\). In fact, we do not know whether the inverse of matrix \(P\) exists or not. However, Theorem 1 guarantees the existence and robustness of the full correlation matrix \(\rho(V_\lambda)\) without specifying the estimation procedure. With the help of Theorem 3, we demonstrate this procedure of step-by-step element-wise estimation running on subvines in algorithm 2. Because the correlation matrix is symmetric, we only care about the upper triangular part \(i < j\).

Algorithm 2: Estimating Robust Correlation Matrix.

\[
\text{Data: partial correlation matrix } P, \text{ semantic vine } V_\lambda \\
\text{Result: correlation matrix } \rho_{u \times u}(V_\lambda) \\
\text{for } i = 1, 2, ..., n \text{ do} \\
\quad \rho_{ii} \leftarrow 1; \\
\text{end} \\
\text{for } (i, j) \in E_1 \text{ and } i < j \text{ do} \\
\quad \rho_{ij} \leftarrow P_{ij}; \\
\text{end} \\
\text{for } k = 2, 3, ..., n - 1 \text{ do} \\
\quad \text{for } (i, j) \in E_k \text{ and } i < j \text{ do} \\
\quad \quad u \leftarrow (i, u) \in E_{k-1}; \\
\quad \quad v \leftarrow (j, v) \in E_{k-1}; \\
\quad \quad \rho_{uv} \leftarrow P_{uv}\left(1 - \rho_{uu}^2\right)\left(1 - \rho_{vv}^2\right) + \rho_{uu}\rho_{vv}; \\
\quad \text{end} \\
\text{end} \\
\text{return } \rho(V_\lambda); \\
\]

In algorithm 2, the asset pairs \((i, u)\) and \((j, v)\) exist and are unique. This is because of the structure of a regular vine. The nodes in \(E_k\) are inherited from \(E_{k-1}\). Assuming there are \((i, u)\) and \((i, u')\) both in \(E_{k-1}, u \neq u'\), then for edge set \(E_k\) we will have \((u, u')j, ..., j \in E_k\) instead of \((i, j) \in E_k\). A similar statement holds for asset \(j\).

Because \((i, u)\) and \((j, v)\) are in \(E_{k-1}\), \(\rho_{uu}\) and \(\rho_{vv}\) are pre-computed with a smaller \(k\). Hence algorithm 2 ensures each \(\rho_{uv}\) is computable and there are \(n + (n - 1) + (n - 2) + \cdots = \frac{n+1}{2}\) assignments. This covers all the unique values required in the correlation matrix.

6. Data description

We investigate a list of 55 stocks from the US markets. A great proportion of this list is the same as used by Zhang [1] and Zhu [9]. Their industry classification codes are retrieved from the Bloomberg Terminal and Thomson Reuters Eikon. Moreover, we obtained the historical closing price of 5 stocks randomly selected from the list from the Quandl API and construct a virtual portfolio. The 5 stocks with their ticker and number are: Apple Inc (1: AAPL), Microsoft Corporation (2: MSFT), Goldman Sachs Group Inc (3: GS), Pfizer Inc (4: PFE), and Wells Fargo & Company (5: WFC). The price data is later applied to the mean-variance optimization (MVO) model of Markowitz [18].

We train the semantic vector space for financial document representation with linguistic materials from two sources: the public available Reuters-21578 Corpus,\(^5\) which contains 10,788 financial news documents totaling 1.3 million words, and the Wikipedia pages of our list of stocks.\(^6\)

5 http://www.davdilewis.com/resources/testcollections/reuters21578.
6 Retrieved from the Internet on 2017-10-09.

7. Experiments

We use the Reuters Company Business Descriptions to generate dense semantic vector representations (100-dimensional) for each stock, the automatically extracted keywords for the selected stocks we used for portfolio construction are listed in Table 1.

<table>
<thead>
<tr>
<th>Stock ticker</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: AAPL</td>
<td>Apple, design, mobile, device, digital, computer, iPhone, software, service, store, application, accessory, support</td>
</tr>
<tr>
<td>2: MSFT</td>
<td>Microsoft, technology, software, business, productivity, develop, system, manufacture, device, computer, solution, intelligent</td>
</tr>
<tr>
<td>3: GS</td>
<td>Goldman, Sachs, investment, bank, management, client, institutional, financial, advisory, security, loan, asset, service</td>
</tr>
<tr>
<td>4: PFE</td>
<td>Pfizer, research, pharmaceutical, healthcare, medicines, vaccines, inflammation, business, generics, consumer, immunology</td>
</tr>
<tr>
<td>5: WFC</td>
<td>Wells Fargo, wholesale, bank, wealth, financial, service, investment, management, commercial, mortgage, retail</td>
</tr>
</tbody>
</table>

We evaluate our method of semantic vine growing and correlation matrix estimation on stock price data by three experiments. The first experiment benchmarks with several portfolio settings without using a vine structure, the second one makes a comparison with a number of arbitrary vine structures. The third experiment addresses the scalability of our method and its application to financial knowledge discovery.

7.1. Semantic vine and asset correlations

We obtain the following semantic vine structure for the selected stocks (Fig. 4) using algorithm 1.

![Image](image.png)

The resulting structure is neither a C-vine nor D-vine. Tree 1 mixes the subvines structure of C-vine and D-vine; Tree 2 has a C-vine structure; Tree 3 and Tree 4 follow a D-vine structure. The logical structure of C-vine and D-vine actually reflects different aspects of the joint distribution of variables. A C-vine is more suitable for dependence modeling when a critical variable “leads” the others, whereas a D-vine is more suitable when the variables are relatively independent. In our semantic vine, Microsoft and Wells Fargo are to some extent “hubs” that bridge other stocks in Tree 1 and Tree 2. Another interpretation is that the least related nodes are joined in the highest order Tree 4, in our case the node numbers 1 and 4 represent Apple and Pfizer, which also have the minimum \(s_i\). In this sense, our semantic vine growing algorithm produces a similar structure to the optimal vine truncation suggested by [30], but with more theoretical soundness.\(^7\)

\(^7\) See Section 5 for the definition of the optimal vine truncation.
We calculate the unconditional full product-moment correlation matrix as a robust correlation matrix estimator for the stocks using algorithm 2:

$$\rho = \begin{bmatrix} 1 & 0.4167 & -0.2267 & -0.6426 & -0.0394 \\ 0.4167 & 1 & 0.3047 & 0.2003 & 0.1240 \\ -0.2267 & 0.3047 & 1 & 0.2909 & 0.0179 \\ -0.6426 & 0.2003 & 0.2909 & 1 & -0.2747 \\ -0.0394 & 0.1240 & 0.0179 & -0.2747 & 1 \end{bmatrix}$$

The daily returns are calculated as:

$$\mu_d = \frac{\text{price}_d - \text{price}_{d-1}}{\text{price}_{d-1}}.$$

Then the portfolio expected return is estimated by averaged $\mu_d$ in a period of length $k$:

$$\bar{\mu} = \frac{1}{k} \sum_{d=1}^{k} \mu_d.$$

We compare five model settings with no short selling, taxes, or transaction fees.

1. The equal-weighted portfolio (EW): we hold the portfolio weight of [0.2, 0.2, 0.2, 0.2, 0.2] throughout the test period. Then the portfolio performance will be an average of the returns from the 5 selected stocks. This strategy is fundamentally yet powerful. Empirical study [32] shows that the equal-weighted portfolio is hard to beat.

2. Robust MVO (rMVO): we use both static covariance matrix estimation and static expected return estimation, which is calculated by averaging returns in the past 30 days. Both estimations are used throughout the test period.

3. MVO (MVO): we use the same static expected return estimation as rMVO, but the covariance is calculated from the return data in the past 90 days.

4. Dynamic robust MVO (drMVO): we use the covariance matrix estimation as rMVO, but the expected return estimation is updated daily on a sliding window of 30 days.

5. Dynamic MVO (dMVO): we use the covariance matrix estimation based on a sliding window of 90 days and the expected return estimation on a sliding window of 30 days.

We test the portfolio trading simulation performance from 2016-03-09 to 2017-09-30 (579 days in total). Fig. 5 gives the capital amount starting from 1 dollar. Table 5 reports the compound annual growth rate (CAGR) and Sharpe ratio [33], which is a popular risk-adjusted return measure among practitioners. The formulas for calculating these two metrics are as follows:

$$\text{CAGR} = \left[\left(\frac{\text{C}_t}{\text{C}_0}\right)^{\frac{365.25}{\text{t}}}-1\right] \times 100\%$$

$$\text{Sharpe ratio} = \frac{\text{E}(\mu_d \cdot \frac{\mu_{pfl}}{\sigma_d})}{\sigma(\mu_{pfl})}$$

where $C$ denotes the capital amount of a portfolio; $\text{E}(\cdot)$ denotes the expectation; $\sigma(\cdot)$ denotes the standard deviation.

**Findings**

In our simulation, rMVO is the only portfolio setting that consistently outperforms the EW. Rest of the portfolios cannot compete with EW even before deducting transaction cost. This result underpins the effectiveness of our robust correlation matrix estimation method with the assistance of a semantic vine structure of assets. MVO exhibits high volatility and bad profitability. This may happen on account of two reasons: the unstable estimation of asset correlations and the static return estimation, which may not be accurate.

Both drMVO and dMVO are not performing as well as EW. This may suggest that the dynamic estimation is not very helpful, as long as it does not involve a robust approach. Moving with similar patterns, dMVO is slightly better than drMVO in terms of expected returns. Though in terms of volatility, dMVO is slightly more stable than dMVO, this difference is insignificant (see Table 2). These observations may suggest the importance of a match of time periods of data used to estimate expected returns and the covariance matrix.

Our experimental results also confirm the statement that the most serious problem of the mean–variance efficient frontier is probably the method’s (in)stability [3]. Although dynamically estimating both expected returns and covariance and daily rebalance, dMVO stays on the elusive efficient frontier, and performs even worse than rMVO.

**7.3. Benchmarking arbitrary vines**

The previous experiment has shown the effectiveness of using our semantic vine to robustly estimate the covariance matrix in the mean–variance optimization construction. However, the quality of this vine structure is unclear. In this experiment, we adopt the rMVO model setting, which has the best performance in the above discussions. We substitute the semantic vine with some random vine structures. Without knowing the dependence structures, random vines are the state-of-the-art modeling that preserves robustness. For the convenience of computing partial correlations, we grow the standard C-vine and D-vine structure and assign random numbers between $-1$ and 1 to each edge. Thus, the edge set for C-vines is $\varepsilon_1 = \{(1, 2), (1, 3), (1, 4), (1, 5)\}$ $\varepsilon_2 = \{(2, 3), (2, 4), (2, 5)\}$ $\varepsilon_3 = \{(3, 4), (3, 5)\}$ $\varepsilon_4 = \{(4, 5)\}$. The edge set for D-vines is $\varepsilon_1 = \{(1, 2), (2, 3), (3, 4), (4, 5)\}$ $\varepsilon_2 = \{(1, 3), (2, 4), (3, 5)\}$ $\varepsilon_3 = \{(1, 4), (2, 5)\}$ $\varepsilon_4 = \{(1, 5)\}$. We obtain 20 alternative vines in total. We use the abbreviated notation, for instance, Cv-n for the nth random C-vine and Dv-n for the nth

<table>
<thead>
<tr>
<th>Portfolio setting</th>
<th>CAGR (%)</th>
<th>Sharpe ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>EW</td>
<td>21.21</td>
<td>1.00</td>
</tr>
<tr>
<td>MVO</td>
<td>-15.90</td>
<td>-0.20</td>
</tr>
<tr>
<td>rMVO</td>
<td>23.68</td>
<td>1.01</td>
</tr>
<tr>
<td>dMVO</td>
<td>16.27</td>
<td>0.78</td>
</tr>
<tr>
<td>drMVO</td>
<td>10.39</td>
<td>0.52</td>
</tr>
</tbody>
</table>
random D-vine. Each portfolio performance is illustrated in Fig. 6, and statistical measures and significance test results are provided in Tables 3 and 4.

**Findings**

The trading simulation results suggest the importance of a proper vine structure. With arbitrarily-grown vines, the portfolio does not always outperform the EW. In fact, Table 3 demonstrates that only 2-out-of −10 C-vines and 1-out-of −10 D-vines provides better portfolio return than the semantic vine in terms of CAGR. However, this is at the expense of significantly higher risk. No arbitrary vine structure outperforms the semantic vine in terms of Sharpe ratio, which means it is not wise to take the risk premium because the compensation is not proportional. The average CAGRs for C-vines and D-vines are −0.1% and 7.2% respectively, both with a large variance and significantly lower than EW. The average Sharpe ratios are 0.30 and 0.28, which is much lower than that of EW and Sv. Significance tests also support that it is hard for an arbitrary vine to outperform the semantic vine. Note that the Chebyshev’s probability bound regardless of the distribution of the random variable is very loose, the probability for such an event in terms of Sharpe ratio is less than 2%.

Furthermore, the robust covariance matrix estimation guarantees robust efficient portfolio weights, but does not necessarily guarantee robust portfolio returns. The portfolio returns of arbitrarily-grown C-vines and D-vines are not stable. Losing all the principal capital, though this did not happen in our simulations, is possible. Vine structures of similar CAGR in Table 3 may have very different Sharpe ratios. This phenomenon means the portfolio return and risk are not tightly related even when a vine structure is used in covariance matrix estimation.

### 7.4. Model scalability

The time complexity of vine growing and robust correlation matrix estimation algorithms is very critical, because a larger number of stocks will generate more complicated vine structure.

We exclude the pre-training time of semantic spaces in our discussion because even if it varies, document vectors can always be embedded in negligible time afterward. However, the scale of the problem also depends on data properties. For example, judging adjacency of edges relates to the average degree of nodes in each layer. In particular, if the judging adjacency process can be done in a quasilinear time, the theoretical complexity for vine-growing would be $O(n^2 \log n)$, while the robust correlation matrix estimation has a theoretical complexity of $O(n^3)$. Considering the naïve calculation of partial correlations has a complexity of $O(n^3)$, the semantic vine can be constructed in an acceptable time. Table 5 reports the CPU time experimented on semantic vine-growing on different numbers of assets and its deviation from estimated time.

Another concern is whether a more complicated semantic vine preserves the quality to reveal the important correlations between assets. We investigate Tree 1 on a larger scale of stocks (Fig. 7). The stocks are colored according to their business sectors suggested by the Global Industry Classification Standard (GICS) and the Thomson Reuters Business Classification (TRBC). The two systems are almost identical in terms of sectors (only 2 out of 55 stocks we experimented are categorized into different sectors). We can observe that many companies in the same industry are linked, such as Wells Fargo and JPMorgan [34]. Whereas these companies can have very different neighbors. For instance, both

![Fig. 5. Performance with different experiment settings.](image-url)
classified as consumer discretionary business, Comcast is more affiliated with telecom industry, while Amazon is located closer to healthcare and retailing business. The first layer dependence structure captures more information than the popular industry classification standards.

8. Conclusion

Diversified approaches are developed to incorporate prior knowledge into financial applications, such as Bayesian fusion. While not many of these approaches attached importance to the issue of robustness. We believe it would be more interesting to leverage prior knowledge with theoretically sound robustness. In this paper, we propose to use the semantic information of financial assets to build a vine dependence structure. We use this semantic vine to tackle the challenging problem of robust correlation matrix estimation in the asset allocation framework. Experiments show that this approach improves portfolio performance. Moreover, the semantic vine is superior to vine structures that are arbitrarily grown without prior knowledge. Since arbitrary vines already help portfolio optimization and knowledge-based vine selection remains rarely-explored, our approach is non-trivial.

Finally, we plan to collect data for a wider range of financial assets in the future. It would be interesting to investigate the vine structure on different types of assets other than company stocks. We also plan to take advantage of textual time series data from social media. Hopefully, this would be beneficial to the discovery of a delicate market structure and the dynamic construction of robust semantic vines.

References