New Statistical Techniques for Designing Future Generation Retirement and Insurance Solutions

by

Zhe Zhu

Bachelor of Mathematics, University of Waterloo (2010)

Submitted to the Sloan School of Management in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2014

Signature redacted

Signature of Author

.....

Sloan School of Management
July 31, 2014

Signature redacted

Certified by.....

.....

Roy E. Welsch
Eastman Kodak Leaders for Global Operations Professor of Management
Professor of Statistics and Engineering Systems
Thesis Supervisor

Signature redacted

Accepted by.....

.....

Dimitris Bertsimas
Boeing Professor of Operations Research
Co-Director, Operations Research Center

New Statistical Techniques for Designing Future Generation Retirement and Insurance Solutions

by

Zhe Zhu

Submitted to the Sloan School of Management
on July 31, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

This thesis presents new statistical techniques for designing future generation retirement and insurance solutions. It addresses two major challenges for retirement and insurance products: asset allocation and policyholder behavior.

In the first part of the thesis, we focus on estimating the covariance matrix for multi-dimensional data, and it is used in the application of asset allocation. Classical sample mean and covariance estimates are very sensitive to outliers, and therefore their robust counterparts are considered to overcome the problem. We propose a new robust covariance estimator using the regular vine dependence structure and pairwise robust partial correlation estimators. The resulting robust covariance estimator delivers high performance for identifying outliers under the Barrow Wheel Benchmark for large high dimensional datasets. Finally, we demonstrate a financial application of active asset allocation using the proposed robust covariance estimator.

In the second part of the thesis, we expand the regular vine robust estimation technique proposed in the first part, and provide a theory and algorithm for selecting the optimal vine structure. Only two special cases of the regular vine structure were discussed in the previous part, but there are many more different types of regular vines that are neither type.

In many applications, restricting our selection to just those two special types is not appropriate, and therefore we propose a vine selection theory based on optimizing the entropy function, as well as an approximation heuristic using the maximum spanning tree to find an appropriate vine structure. Finally, we demonstrate the idea with two financial applications.

In the third part of the thesis, we focus on the policyholder behavior modeling for insurance and retirement products. In particular, we choose the variable annuity product, which has many desirable features for retirement saving purposes, such as stock-linked growth potential and protection against losses in the investment. Policyholder behavior is one of the most important profit or loss factors for the variable annuity product, and insurance companies generally do not have sophisticated models at the current time. We discuss a few new approaches using modern statistical learning techniques to model policyholder withdrawal behavior, and the result is promising.

Thesis Supervisor: Roy E. Welsch
Title: Eastman Kodak Leaders for Global Operations Professor of Management
Professor of Statistics and Engineering Systems

Acknowledgements

I am very grateful to my thesis supervisor Professor Roy E. Welsch for his continuous guidance and support throughout my entire PhD study. It is a great privilege to be able to work with him. Professor Welsch introduced me to the world of robust statistics and the importance and usefulness of such techniques, especially in the insurance and financial applications. It is a great pleasure working with him, and such experience will have a great influence on my future research career.

I would also like to thank my thesis committee members, Professor Leonid Kogan and Professor Patrick Jaillet, for their support through the years. I truly appreciate their thoughtful comments and suggestions for my thesis research, both the theoretical parts and the applicational ones.

In addition, I would like to thank my family for their love and sacrifices over my entire study. Also, my colleagues at the Operations Research Center and other related departments for their advices and supports, especially Yehua Wei for helping me throughout my first several years at MIT, and for his friendship over the years.

Finally, this research is supported in part by the Society of Actuaries (SOA), the Natural Sciences and Engineering Research Council of Canada (NSERC), the MIT Center for Computational Research in Economics and Management Science, and the Singapore MIT Alliance (SMA2).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

This thesis introduces new statistical techniques for designing future generation retirement and insurance solutions. The current designs for retirement and insurance products are facing challenges from both the investment side and the policyholder behavior side, and the recent financial crisis of 2008 has exacerbated the situation for insurance companies and pension funds with significantly shrinking asset values and unexpected policyholder withdrawals. The advances in modern statistical modelling techniques will enable these entities to better understand, design and manage the risk of their products. In the rest of this chapter, we provide a brief motivation and description of the specific techniques that we have developed in this thesis, and then we provide the organizational structure of the rest of the chapters.

1.2 Research Motivations

In this section, we discuss the motivations for this research topic and the various challenges presented in the current literature.

After the 2008 financial crisis, with the almost-failure of the insurance giant AIG and many pension plans, it is of critical importance to study and design future generation retirement and insurance solutions that are more quantifiable in risk modeling. There are three major features for any retirement and insurance solution. First, they will have a central accumulation piece, mostly a mixed investment in stocks and bonds. Second, they

will have guarantee options, such as a minimum accumulation or withdrawal guarantees, and these guarantees will be different from financial options because they are not tradable and very long term. Lastly, they will experience uncertain policyholder behaviors, which arise due to personal circumstances and are not necessarily rational in the sense of an institutional investor. Since the second feature has been studied extensively by the finance community, this thesis will focus on the first and third feature.

The advancements in modern statistics allow us to tackle some of these challenges with better results compared to existing practices. For example, the asset allocation problem requires a robust covariance estimator that can identify and filter out outliers/extreme observations in the financial data, and the techniques in the robust statistics literature present promising models for such a problem. Another example is that the advancement of modern statistical learning methods allows building more sophisticated models and making more accurate predictions for policyholder behaviors.

1.3 The Structure of the Thesis

In Chapter 2 of the thesis, we introduce a new robust dependence modeling technique for high-dimensional covariance matrices. Estimating the covariance matrix is a fundamental step in the asset allocation application, and it is well known that classical sample covariance estimator is very sensitive to outliers. With a distorted covariance estimate, a portfolio optimizer may give unrealistically extreme weights to certain stock. In this chapter, we address this challenge by first introducing the concept of a regular vine, which is a model for dependence structure and the various special cases of the regular vines. Then, we introduce the partial correlation vine by incorporating partial correlations to any given regular vine structure. Additionally, we add robust estimation to the process to make the resulting correlation/covariance matrix less sensitive to extreme observations. Such an estimation process provides many nice theoretical properties such as the preservation of breakdown and guaranteed positive-definiteness of the robust correlation/covariance

matrix. The resulting robust covariance estimator delivers high performance for identifying outliers under the Barrow Wheel Benchmark for large high dimensional datasets. Finally, we demonstrate a financial application of active asset allocation using the proposed robust covariance estimator, and the proposed estimator delivers better results compared to many existing asset allocation methods.

In Chapter 3 of the thesis, we extend the research from Chapter 2 and propose a systematic method for the vine selection process. This chapter solves this challenge by selecting the vine that produces the least entropy from the resulting covariance/correlation estimate. However, the number of regular vines grows exponentially as the number of variables increases, and therefore we provide a heuristic search algorithm that uses a greedy method to find the optimal vine level by level. In this chapter, we also provide some nice properties (with proofs) of the selected vine, such as preservation of breakdown, permutation invariance and favorable optimality gap via simulation. Finally, we use the method on two different applications: 1) asset allocation and 2) uncovering of the underlying dependence structure of different industries.

In Chapter 4 of the thesis, we shift the focus to the modeling of policyholder behavior, another important perspective that deserves study. Unlike institutional investors who monitor indices like credit ratings and risk sensitivities of a portfolio, insurance and annuity policyholders choose to withdraw their contract based on personal circumstances such as the need for financing. Modern statistical learning techniques provide a great foundation for modelling such behavior, while the insurance and pension industries have not yet fully explored those techniques. This chapter first applies the traditional logistic regression technique, and uses the result as the baseline. Then, we apply varies modern statistical learning techniques, and determine the best model according to 10-fold cross validation. The final result is that the best model, which is Rotation Forest, performs better than the baseline logistic model statistically in terms of accuracy.

1.4 The Contributions of the Thesis

This thesis makes the following theoretical and applicational contributions:

1. Introduces a new robust dependence/correlation modeling technique that incorporates both the regular vine structure and robust partial correlation modeling. Such technique guarantees the positive definiteness of the estimated correlation matrix, preserves the breakdown point in the estimation process, and delivers good results in the benchmark while improving computing time for large datasets.

2. Formulates the optimization problem for finding the optimal robust vine based on minimum entropy, and proposes a heuristic to solve this optimization problem by going through the vine level by level, and finds the maximum spanning tree according to the robust correlation measure on each level.

3. Applies the proposed technique to the asset allocation problem with investment assumptions suitable for insurance companies, and the allocation method using the covariance matrix estimated by this new technique outperforms other popular asset allocation methods based on both accumulation and the information ratio.

4. Applies varies modern statistical learning techniques to the application of policyholder withdrawal behavior for the variable annuity product. The best statistical learning method, which is selected according to 10-fold cross validation, performs better statistically compared to the traditional logistic regression method.

Chapter 2

Robust Dependence Modeling for High-Dimensional Covariance Matrices

In this chapter, we introduce the vine dependence modeling structure and robust partial correlation estimation, and combine both techniques to model high-dimensional covariance matrices.

2.1. Introduction

In multivariate analysis, estimating the dispersion/covariance matrix is a fundamental step for many applications, such as active asset allocation and credit risk dependence modelling. Classical sample covariance estimates are very sensitive to outliers, and therefore their robust counterparts are considered to address this problem. However, there are many challenges in the literature of robust covariance estimation for high dimensional datasets, which include low breakdown point (i.e., the maximum percentage of contaminations that can be tolerated), computational inefficiency, and no guarantee of positive definiteness. Also, in many applications, there may be prior knowledge about the data dependence structure that can be used in the modelling. For example, in the asset allocation problem, we know which industry each stock falls under, and we should be able to use such knowledge when estimating the covariance matrix.

2.2. Literature Review

There are many existing robust procedures for estimating covariance matrices for multivariate datasets, but they suffer from one or more of the shortcomings mentioned in the previous section.

Multivariate M-estimators use generalized maximum likelihood estimators (MLEs) when estimating covariance matrices. M estimators are computationally efficient in general. However, for monotone M estimators, the breakdown point is $1/(1+p)$, where p is the number of variables. Tyler (1987) showed an M estimator with breakdown being $1/p$, which is a bit higher, but with p being really big, such an estimator does not make a difference.

Later, multivariate covariance estimators with high breakdown point were introduced. For example, minimum covariance determinant (MCD) and minimum volume ellipsoid (MVE) were presented in Rousseeuw and Leroy (1987). These estimators have many attractive properties, such as high breakdown that can be specified, guaranteed positive definiteness of the covariance estimate, and affine-equivariance. However, these methods require heavy computation. For example, MCD was not popular until the introduction of Fast-MCD (Rousseeuw and Driessen, 1999), but it is still computationally heavy compared to estimators like M-estimators, and it may get stuck in a local minimum. With a big data set with many variables, MCD may not be computationally feasible.

Finally, there have been discussions of using robust pair-wise covariances as entries for the covariance matrix. Since there are many computationally-efficient robust procedures for estimating pair-wise covariances, the estimation of the covariance matrix is also computationally efficient. However, the resulting covariance matrix may not be positive definite, and this violates the fundamental property of a covariance matrix. Many papers, such as Maronna and Zamar (2002), have investigated techniques to ensure positive-definiteness.

This chapter will introduce a new robust covariance estimator using a dependence structure called a vine. Together with high-breakdown robust regression estimators to compute partial correlations, our proposed method achieves a high breakdown point, guaranteed positive-definiteness of the covariance matrix, as well as faster run time compared to MCD for datasets with lots of data points.

2.3. Vine Dependence Structure

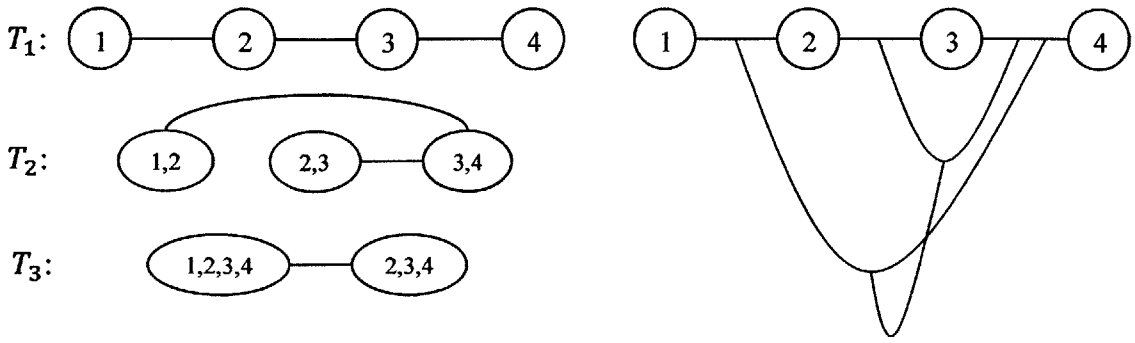
A vine is a graphical tool for modeling dependence structure in high dimensions, and it was introduced in Bedford and Cooke (2002) and Kurowicka and Cooke (2006). A regular vine is a special case where dependence structures are specified for two variables conditional on some other variables. Regular vines generalize tree structures, and combined with robust statistics techniques, they have proven to be a valuable tool in high-dimensional robust modeling.

Definition 2.3.1 (Vine). $\mathcal{V}$ is a vine on p elements with $\mathcal{E}(\mathcal{V}) = \mathcal{E}_1 \cup \ldots \cup \mathcal{E}_{p-1}$ denoting the set of edges of $\mathcal{V}$ if

1. $\mathcal{V} = \{T_1, \ldots, T_{p-1}\}$;
2. T_1 is a connected tree with nodes $N_1 = \{1, \ldots, p\}$, and edges $\mathcal{E}_1$;
3. For $i = 2, \ldots, p-1$, T_i is a tree with nodes $N_i = E_{i-1}$.

There are usually two graphical ways to represent a vine structure. Figure 2.3.1 shows the two graphical representations of the same vine. The graph on the left follows directly from Definition 2.3.1, and it shows all the trees T_i. The graph on the right is a more compact way of representing the vine structure. It combines all the trees T_i by connecting the edges from the previous level.

Figure 2.3.1 - Graphical Representations of a Vine



Definition 2.3.2 (Regular Vine). $\mathcal{V}$ is a regular vine on p elements with $\mathcal{E}(\mathcal{V}) = \mathcal{E}_1 \cup \ldots \cup \mathcal{E}_{p-1}$ denoting the set of edges of $\mathcal{V}$ if

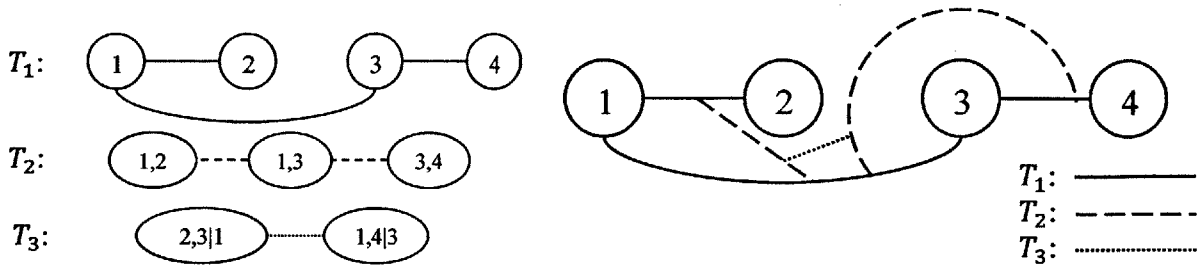
1. $\mathcal{V}$ is a vine;
2. (**proximity**) For $i = 2, \ldots, p-1$, $\{a, b\} \in \mathcal{E}_i$, $\#(a \Delta b) = 2$ where Δ denotes the symmetric difference operator, and # denotes the cardinality of a set
    - $a \Delta b = (a \cup b) \backslash (a \cap b)$

On a regular vine, each edge connecting the nodes a and b corresponds to the dependence structure of the pair $a \Delta b$ conditional on $a \cap b$. The proximity property guarantees that $a \Delta b$ always has exactly two variables.

Following Definition 2.3.2, the vine in Figure 2.3.1 is not a regular vine because it violates the proximity condition. Specifically, at T_2, $\{1,2\} \Delta \{3,4\} = \{1,2,3,4\}$, so $\#(\{1,2\} \Delta \{3,4\}) = 4 \neq 2$.

Figure 2.3.2 shows a sample regular vine with the two graphical representations. This sample regular vine is neither a C-vine or a D-vine, which are to be defined soon.

Figure 2.3.2 - Graphical Representations of a Regular Vine

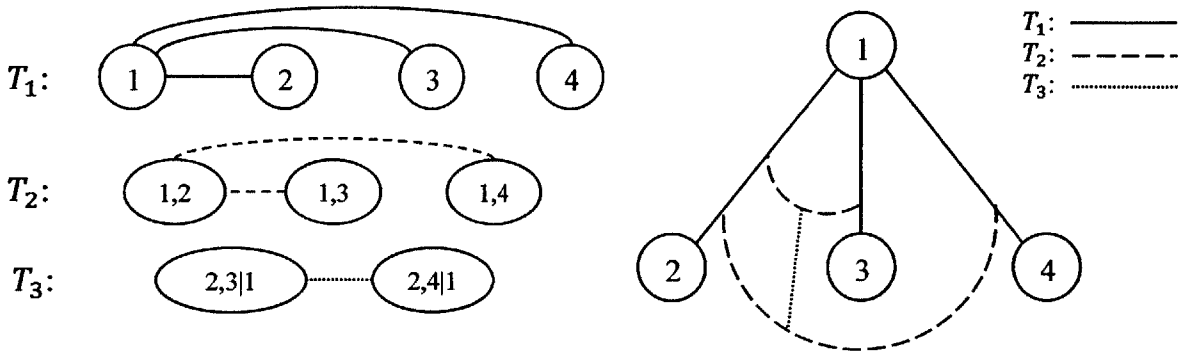


There are two special types of regular vine, namely C-vine and D-vine. They have very specific structures, and they are very useful as initial dependence structure models when there is not much prior knowledge about the dependence structure.

Definition 2.3.3 (C-vine). A regular vine is called a **Canonical** or **C-vine** if each tree T_i has a unique node of degree $p - i$. The node with maximal degree in T_i is the **root**.

Figure 2.3.3 is a sample C-vine on 4 variables. The roots are {1}, {1,2} and {2,3|1} for T_1, T_2 and T_3 respectively.

Figure 2.3.3 - Example of a C-vine



Definition 2.3.4 (D-vine). A regular vine is called a **Drawable** or **D-vine** if each node in T_1 has a degree of at most 2.

Figure 2.3.4 - Example of a D-vine

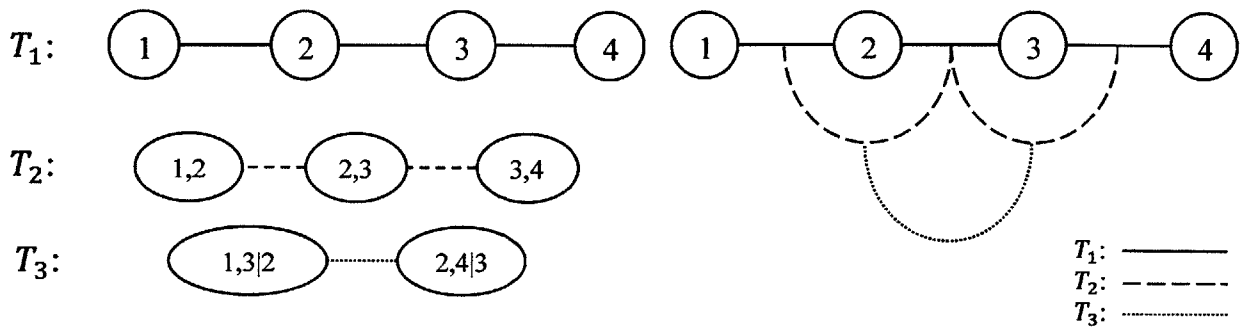
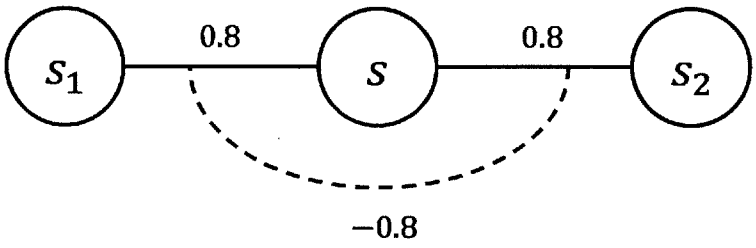


Figure 2.3.4 is a sample D-vine on 4 variables. Each tree is a path, so it is the most drawable vine.

Here is a financial example of using the regular vine structure. Let s_1 and s_2 be the stock returns of two competing companies in the same industry, such as Apple and Google. Let s be the stock return for that industry sector, in this case the IT industry.

Figure 2.3.5 gives a possible regular vine dependence structure for s_1, s_2 and s, and the pairwise correlations and partial correlation are displayed on the edges. For example, the correlation between s_1 and s is 0.8, which shows that the return of stock 1 is positively correlated to the industry stock return, and this assumption is reasonable. Similarly, the return of stock 2 is positively correlated to the industry stock return as well. Finally, the partial correlation between s_1 and s_2 given s is -0.8, which shows that the return of stock 1 is negatively correlated to the return of stock 2 while controlling for the industry stock return (s). This reflects that the two companies are competitors within their industry.

Figure 2.3.5 - Financial Example of a Regular Vine



2.4. Robust Correlation Estimation

In this section, we introduce a new robust correlation estimation method using pairwise partial correlations. We first provide the definition of partial correlation, and then describe the partial correlation vine structure and its relationship with the correlation matrix. Finally, we incorporate a robust estimation technique for partial correlations and use it to construct the robust correlation matrix that guarantees positive-definiteness.

Definition 2.4.1 (Partial correlation). Consider random variables $X_1, \dots, X_n$.
Let $X^*_{1;3,\dots,n}$ and $X^*_{2;3,\dots,n}$ be the best linear approximations to X_1 and X_2 based on the variables $X_3, \dots, X_n$. Let $Y_1 = X_1 - X^*_{1;3,\dots,n}$, $Y_2 = X_2 - X^*_{2;3,\dots,n}$ be the residuals. Then, the partial correlation between X_1 and X_2 given all other variables, denoted by $\rho_{1,2;3,\dots,n}$ is defined as the ordinary correlation coefficient between Y_1 and Y_2.

Therefore, the partial correlation $\rho_{1,2;3,\dots,n}$ can be interpreted as the correlation between the orthogonal projections of X_1 and X_2 on the plane orthogonal to the space spanned by $X_3, \dots, X_n$. Partial correlations can be computed from correlations using the following recursive formula (Yules and Kendall, 1965):

$$\rho_{1,2;3,\ldots n} = \frac{\rho_{1,2;3,\ldots,n-1} - \rho_{1,n;3,\ldots,n-1} \cdot \rho_{2,n;3,\ldots,n-1}}{\sqrt{1-\rho^2_{1,n;3,\ldots,n-1}} \cdot \sqrt{1-\rho^2_{2,n;3,\ldots,n-1}}} \tag{2.4.1}$$

Also, the partial correlation can be computed directly from correlation matrix, Σ. Define $P = \Sigma^{-1}$, then:

$$\rho_{i,j;\{1,\ldots n\}\backslash\{i,j\}} = -\frac{p_{ij}}{\sqrt{p_{ii}p_{jj}}} \tag{2.4.2}$$

Where p_{ij} is the $(i,j)^{\text{th}}$ entry of P.

Theorem 2.4.1 (Bedford and Cooke, 2002). For any regular vine on p elements, there is a one-to-one correspondence between the set of $p \times p$ positive definite correlation matrices and the set of partial correlation specifications for the vine.

As will be shown in the proof for Theorem 2.4.2, a positive definite correlation matrix cannot have an off-diagonal entry equal to -1 or 1. Therefore, by Theorem 2.4.1, any assignment of the numbers strictly between -1 and 1 to the edges of a partial correlation regular vine is consistent with a positive definite correlation matrix, and all positive definite correlation matrices can be obtained this way.

It can be verified that the correlation between the i^{th} and j^{th} variables can be computed from the sub-vine generated by the constraint set of the edge whose conditioned set is $\{i,j\}$. The detailed proof is in the Bedford and Cooke (2002) paper.

Definition 2.4.2 (Partial correlation vine). A partial correlation vine is obtained by assigning a partial correlation ρ_ε, with a value chosen arbitrarily in the interval $(-1,1)$, to each edge ε in $\mathcal{E}(\mathcal{V})$ of the vine defined in Definition 2.3.1. Theorem 2.4.1 shows that there is a bijection between regular partial correlation vines and (positive definite) correlation matrices.

As mentioned previously, in robust covariance estimation, having a high breakdown point is a favorable, yet difficult, property for a robust estimator. We will first provide a formal definition of the breakdown point, and then we will prove that with high-breakdown robust partial correlation estimators, which are readily available, we can produce robust covariance estimators with the same high breakdown as those robust partial correlation estimators.

Definition 2.4.3 (Breakdown point or BP). Let θ be the parameter of interest (can be a vector) that ranges over a set Θ. Let $\hat{\theta} = \hat{\theta}(\boldsymbol{x})$ be an estimate defined for samples $\boldsymbol{x} = \{x_1, \ldots, x_n\}$. The replacement finite-sample breakdown point of $\hat{\theta}$ at $\boldsymbol{x}$ is the largest proportion ϵ^* of data points that can be arbitrarily replaced by outliers without $\hat{\theta}$ leaving a set which is bounded and also bounded away from the boundary of Θ (Donoho and Huber, 1983). In other words, there exists a closed and bounded set $K \subset \Theta$ such that $K \cap \partial\Theta = \emptyset$ (where $\partial\Theta$ denotes the boundary of Θ), and for any $\epsilon \leq \epsilon^*$ contamination of data, $\hat{\theta}$ remains in the set K.

Intuitively, BP measures the maximum amount of data that can be contaminated without the estimator being completely useless. Now, we will show that the BP for the robust correlation estimator is preserved using a partial correlation vine with a robust partial correlation estimator on each edge of the vine.

Theorem 2.4.2. Given a partial correlation vine structure $\mathcal{V}$, we apply robust partial correlation estimation on each edge ε in $\mathcal{E}(\mathcal{V})$ of the vine, and estimate the correlation matrix from the partial correlation estimates using Theorem 2.4.1. If each robust partial correlation estimator has BP at least ϵ^*, then the resulting correlation matrix also has BP at least ϵ^*.

Proof:

There are $\binom{p}{2}$ correlations to be estimated, and each ranges from -1 to 1. Therefore, the parameter space for correlations is $\Theta = [-1,1]^{\binom{p}{2}}$. Also, for any regular vine, there are $\binom{p}{2}$ number of edges (Kurowicka, Cooke and Callies, 2006), and therefore, the parameter space for partial correlations on the vine is $\widetilde{\Theta} = [-1,1]^{\binom{p}{2}}$. For each edge ε of the partial correlation vine, let $\widetilde{\Theta}_\varepsilon = [-1,1]$ be the parameter space for the partial correlation on that edge.

First we will show that, given a partial correlation vine structure $\mathcal{V}$, there exists a continuous function mapping from the correlation matrix to the partial correlations on the edges $\mathcal{E}(\mathcal{V})$ of the vine.

According to Formula 2.4.2 and its generalization, if Σ is the correlation matrix, and let $P = \Sigma^{-1}$, then $\rho_{i,j;\{1,\dots n\}\{i,j\}} = -\frac{p_{ij}}{\sqrt{p_{ii}p_{jj}}}$. Because both the matrix inversion and the computation of partial correlation from the inverse matrix are continuous mappings, there exists a continuous function mapping, denoted by f, from the correlation matrix to the partial correlations on the edges $\mathcal{E}(\mathcal{V})$.

Now, since each robust partial correlation estimator has BP at least ϵ^*, by Definition 2.4.3 (BP), for each robust partial correlation estimator $\tilde{\theta}_\varepsilon$ on the edge ε, there exists a closed and bounded set $\widetilde{K}_\varepsilon \subset \widetilde{\Theta}_\varepsilon$ such that $\widetilde{K}_\varepsilon \cap \partial\widetilde{\Theta}_\varepsilon = \emptyset$, and for any $\epsilon \leq \epsilon^*$ contamination of data, $\tilde{\theta}_\varepsilon$ remains in the set $\widetilde{K}_\varepsilon$. In this case, $\widetilde{\Theta}_\varepsilon = [-1,1]$, and $\partial\widetilde{\Theta}_\varepsilon = \{-1,1\}$, so $-1 \notin \widetilde{K}_\varepsilon$ and $1 \notin \widetilde{K}_\varepsilon$.

Let $\widetilde{K} = \prod_\varepsilon \widetilde{K}_\varepsilon$, the Cartesian product of the sets. Let $K = f^{-1}(\widetilde{K})$, and without loss of generality (WLOG), say $K \subseteq \widetilde{\Theta}$, and if not, we can take $K = f^{-1}(\widetilde{K}) \cap \widetilde{\Theta}$, and the argument will still follow. Now, since the inverse image of a closed set under continuous mapping is also closed (Rudin, 1976), K is closed. Note that $\hat{\theta} \in K$.

Finally, we have to prove that $K \cap \partial\Theta = \emptyset$, and we will prove this by showing that none of the off-diagonal elements of the correlation matrix can be -1 or 1. Suppose, to reach a contradiction, that there is an off-diagonal element of the correlation matrix that is -1 or 1. Since we can reorder the variables, WLOG, let's say such element is ρ_{12}. Therefore, the determinant of the upper left 2-by-2 corner is:

$$\det\begin{bmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{bmatrix} = 1 - \rho_{12}^2 = 0$$

By Sylvester's criterion, the correlation matrix is not positive definite. However, this violates Theorem 2.4.1. Therefore, we can conclude that none of the off-diagonal elements of the correlation matrix can be -1 or 1, and hence $K \cap \partial\Theta = \emptyset$. ■

Theorem 2.4.2 shows that BP of the final robust correlation matrix estimator is inherited from the robust partial correlation estimators. There are many robust partial correlation estimators with high BP, and the resulting robust correlation matrix estimator will have high BP as well.

Now, we will propose a robust estimation procedure for computing robust covariance/correlation matrices following the principles described above.

Theorem 2.4.3 (Yule and Kendall, 1965). Consider variable X_i with zero mean, $i = 1, \dots, n$. Let the numbers $b_{i,j;\{1,\dots n\}\backslash\{i,j\}}$ minimize

$$E\left[\left(X_i - \sum_{j:j\neq i} b_{i,j;\{1,\dots,n\}\backslash\{i,j\}} X_j\right)^2\right], i = 1, \dots, n$$

Then, the partial correlation can be computed as

$$\rho_{i,j;\{1,\dots,n\}\backslash\{i,j\}} = sgn\left(b_{i,j;\{1,\dots,n\}\backslash\{i,j\}}\right)\left|b_{i,j;\{1,\dots,n\}\backslash\{i,j\}} b_{j,i;\{1,\dots,n\}\backslash\{i,j\}}\right|^{\frac{1}{2}}$$

We apply robust regression estimators, such as MM estimator with high breakdown and efficiency, to estimate the robust partial correlations on the vine. Then according to Theorem 2.4.1, we can estimate the correlation matrix robustly using these robust partial correlation estimates. These robust regression estimators are generally faster to compute compared to MCD, and we will demonstrate this in the Benchmark section. Finally, the robust covariance matrix is constructed using the robust correlation matrix and the robust covariance for each individual variable.

Theorem 2.4.1 guarantees the construction of a correlation matrix from partial correlation vines, but there is no algorithm specified how to do so in real-life applications. There are special regular vine structures where algorithms can be specified for such a process, and we will provide algorithms for both the C-vine and D-vine.

For C-vine, the partial correlations are of the form $\rho_{i,j;1,\dots,i-1}$, where $i < j$.

From Formula 2.4.1, we have $\forall k < i$

$$\rho_{i,j;1,\dots,k} = \frac{\rho_{i,j;1,\dots,k-1} - \rho_{k,i;1,\dots,k-1} \cdot \rho_{k,j;1,\dots,k-1}}{\sqrt{1-\rho^2_{k,i;1,\dots,k-1}} \cdot \sqrt{1-\rho^2_{k,j;1,\dots,k-1}}} \tag{2.4.3}$$

Rearrange terms to get a recursive formula for $\rho_{i,j;1,\dots,k-1}$:

$$\rho_{i,j;1,\dots,k-1} = \rho_{i,j;1,\dots,k} \cdot \sqrt{1-\rho^2_{k,i;1,\dots,k-1}} \cdot \sqrt{1-\rho^2_{k,j;1,\dots,k-1}} + \rho_{k,i;1,\dots,k-1} \cdot \rho_{k,j;1,\dots,k-1} \tag{2.4.4}$$

Therefore, for $k = i-1, i-2, \dots, 1$, we can recursively compute $\rho_{i,j;1,\dots,k-1}$, and when $k = 1$, $\rho_{i,j} = \rho_{i,j;1,\dots,k-1}$.

This algorithm has $O(p^3)$ runtime.

Here is an example how this is done for $p = 4$. For a C-vine with 4 variables, robust partial correlation estimators give estimates for the following partial correlations: $\rho_{1,2}$, $\rho_{1,3}$, $\rho_{1,4}$, $\rho_{2,3;1}$, $\rho_{2,4;1}$ and $\rho_{3,4;1,2}$. We can compute the remaining correlations with the following formulas:

$$\rho_{2,3} = \rho_{2,3;1}\sqrt{(1-\rho_{1,2}^2)(1-\rho_{1,3}^2)} + \rho_{1,2}\rho_{1,3}$$

$$\rho_{2,4} = \rho_{2,4;1}\sqrt{(1-\rho_{1,2}^2)(1-\rho_{1,4}^2)} + \rho_{1,2}\rho_{1,4}$$

$$\rho_{3,4;1} = \rho_{3,4;1,2}\sqrt{(1-\rho_{2,3;1}^2)(1-\rho_{2,4;1}^2)} + \rho_{2,3;1}\rho_{2,4;1}$$

$$\rho_{3,4} = \rho_{3,4;1}\sqrt{(1-\rho_{1,3}^2)(1-\rho_{1,4}^2)} + \rho_{1,3}\rho_{1,4}$$

For D-vine, the partial correlations are of the form $\rho_{i,j;i+1,\dots,j-1}$, where $i < j$.

By Anderson (1958)

$$\rho_{i,j} = r_1'(i,j)R_2(i,j)^{-1}r_3(i,j) + \rho_{i,j;i+1,\dots,j-1}D_{i,j} \qquad (2.4.5)$$

Where

- $r_1'(i,j) = (\rho_{i,i+1}, \dots, \rho_{i,j-1})$
- $r_3'(i,j) = (\rho_{j,i+1}, \dots, \rho_{j,j-1})$
- $R_2(i,j) = \begin{pmatrix} 1 & \rho_{i+1,i+2} & \cdots & \rho_{i+1,j-1} \\ \rho_{i+2,i+1} & 1 & & \rho_{i+2,j-1} \\ \vdots & & \ddots & \vdots \\ \rho_{j-1,i+1} & \rho_{j-1,i+2} & \cdots & 1 \end{pmatrix}$
- $D_{i,j}^2 = \left(1 - r_1'(i,j)R_2(i,j)^{-1}r_1(i,j)\right)\left(1 - r_3'(i,j)R_2(i,j)^{-1}r_3(i,j)\right)$

We compute $\rho_{i,j}$ in the following order: i steps up from 3 to p, and for each i, j steps down from $i-2$ to 1.

Here is an example how this is done for $p = 4$. For a D-vine with 4 variables, robust partial correlation estimators give estimates for the following partial correlations: $\rho_{1,2}$, $\rho_{2,3}$, $\rho_{3,4}$, $\rho_{1,3;2}$, $\rho_{2,4;3}$ and $\rho_{1,4;2,3}$. We can compute the remaining correlations in the order: $\rho_{1,3}$, $\rho_{2,4}$, $\rho_{1,4}$.

We can first calculate $\rho_{1,3}$ because

- $r_1'(1,3) = (\rho_{1,2})$ is known;
- $r_3'(i,j) = (\rho_{2,3})$ is known;
- $R_2(i,j) = (1)$ is known;
- $\rho_{1,3;2}$ is known.

Using Formula 2.4.5, we can compute $\rho_{1,3}$.
Similarly, we can then calculate $\rho_{2,4}$.

Finally, we can calculate $\rho_{1,4}$ because

- $r_1'(1,4) = (\rho_{1,2}, \rho_{1,3})$ is known;
- $r_3'(1,4) = (\rho_{2,4}, \rho_{3,4})$ is known;
- $R_2(1,4) = \begin{pmatrix} 1 & \rho_{2,3} \\ \rho_{2,3} & 1 \end{pmatrix}$ is known;
- $\rho_{1,4;2,3}$ is known.

Matrix inversion operations, each with $O(p^3)$ runtime, are required, so naively implementing an algorithm for D-vine requires $O(p^5)$ runtime. However, one can cleverly use block matrix inversion techniques to improve the runtime to $O(p^4)$.

2.5. Benchmark

There have been many proposals made to estimate covariance matrices robustly. Some authors in the field of robust estimation have emphasized the importance of high breakdown point, and Theorem 2.4.2 guarantees that our estimator can achieve a high breakdown point. Others have emphasized speed for large p (Maechler and Stahel, 2009), and our method has an $O(p^3)$ runtime using a C-vine. However, there is usually one important criterion that is overlooked when evaluating a robust covariance estimator, namely the effectiveness of identifying the outlier part. The Barrow Wheel Benchmark (Maechler and Stahel, 2009) has been proposed as a benchmark to evaluate such a criterion.

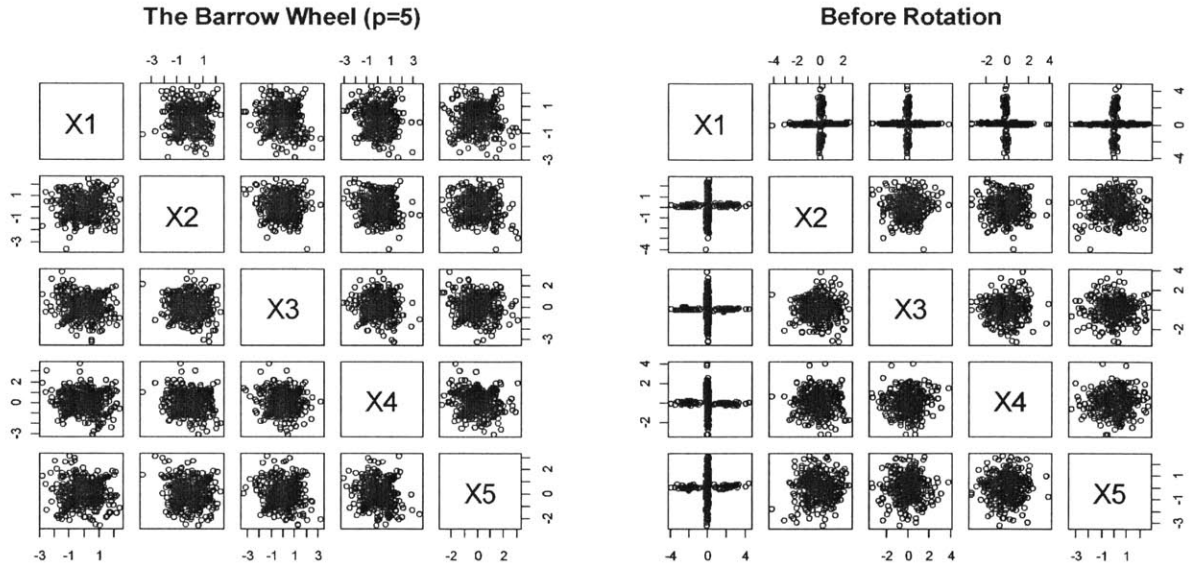
Definition 2.5.1 (Barrow Wheel). The "Barrow Wheel" distribution is a mixture of a flat normal distribution contaminated with a portion $\epsilon = 1/p$ (p is the dimension) of gross errors concentrated near a one-dimensional subspace. Formally, Let

$$G_0 = (1-\epsilon)\cdot \mathcal{N}_p\big(\mathbf{0}, \mathrm{diag}(\sigma_1^2, 1, \ldots, 1)\big) + \epsilon \cdot H$$

with H being the distribution of $\boldsymbol{Y}$, where $Y^{(1)}$ has a symmetric distribution with $\big(Y^{(1)}\big)^2 \sim \chi^2_{p-1}$ and is independent of $Y^{(2)}, \ldots, Y^{(p)} \sim \mathcal{N}_p\big(\mathbf{0}, \sigma_2^2 I_{p-1}\big)$. Then, this distribution is rotated such that the $X^{(1)}$ axis points in the space diagonal direction $(1,1,\ldots,1)$, and the components are rescaled to obtain G, which is the Barrow Wheel distribution.

Figure 2.5.1 shows the data generated by the "Barrow Wheel" distribution with $p = 5$, $n = 400$, $\sigma_1 = 0.05$ and $\sigma_2 = 0.1$, where n is the number of data points, and p is the number of variables.

Figure 2.5.1 - Pairwise Scatter Plot for the Barrow Wheel Distribution (p=5)



The "good" portion of data is $\mathcal{N}_p\big(\mathbf{0}, \mathrm{diag}(\sigma_1^2, 1, \ldots, 1)\big)$, i.e. the wheel, and the outlier portion is H, i.e. the axle. An effective robust estimator should be able capture the wheel and not the axle. We will use the concept of condition number or kappa to measure a robust estimator's ability to identify and isolate outliers.

Definition 2.5.2 (Condition number or kappa). For any positive definite matrix S, let $\lambda_{max}(S)$ and $\lambda_{min}(S)$ be the largest and smallest eigenvalues of S respectively. Then, the condition number or kappa of S is defined as:

$$\kappa(S) = \frac{\lambda_{max}(S)}{\lambda_{min}(S)}$$

With the convention of setting $\sigma_1 < 1$, the theoretical condition number of the covariance matrix for the wheel should be $1/\sigma_1^2$. An effective robust estimator should capture most of the data in the wheel, and hence the condition number of the estimated covariance matrix should be close to the theoretical value of $1/\sigma_1^2$.

As an experiment, we set $\sigma_1 = 0.05$ and $\sigma_2 = 0.1$ for the remainder of this section, and therefore, the theoretical condition number of the covariance matrix for the wheel is $1/0.05^2 = 400$. Maechler and Stahel (2009) provided box-plots (Figure 2.5.2) of condition numbers for a list of robust covariance estimators, and they demonstrated that many cheap "robust" estimators, such as BACON (Billor, Hadi and Velleman, 2000), are as ineffective as the classical covariance estimator with condition number very close to 1. Also, methods like MCD and MVE perform really well, but they are computationally time-consuming. OGK (Maronna and Zamar, 2002) methods perform better than those cheap "robust" methods, but not by a significant margin.

We performed the Barrow Wheel benchmark using our proposed robust covariance estimator. We used C-vine, D-vine and the optimally selected vine as the regular vine structures, and we used both the MM-estimator and Least Trimmed Squares (LTS) as the robust regression estimator for the partial correlations. To make the result comparable with the ones in Figure 2.5.2, we used the same parameters: $n = 100, p = 5, n_{sim} = 50$. We also included the "Oracle" estimator, which gives the estimation of the covariance matrix knowing which portion is the wheel part. The box-plots in Figure 2.5.3 show that our estimators perform much better than many robust covariance estimators in Figure 2.5.2. For example, the OGK estimator in Figure 2.5.2, which uses pair-wise estimation techniques for constructing the robust covariance, has a median condition number well below 100, and our proposed estimators, as shown in Figure 2.5.3, all have condition number above 200. The D-Vine (LTS) estimator even has similar performance to the Oracle one.

Figure 2.5.2 - Benchmark Results from Maechler and Stahel (2009)'s Talk

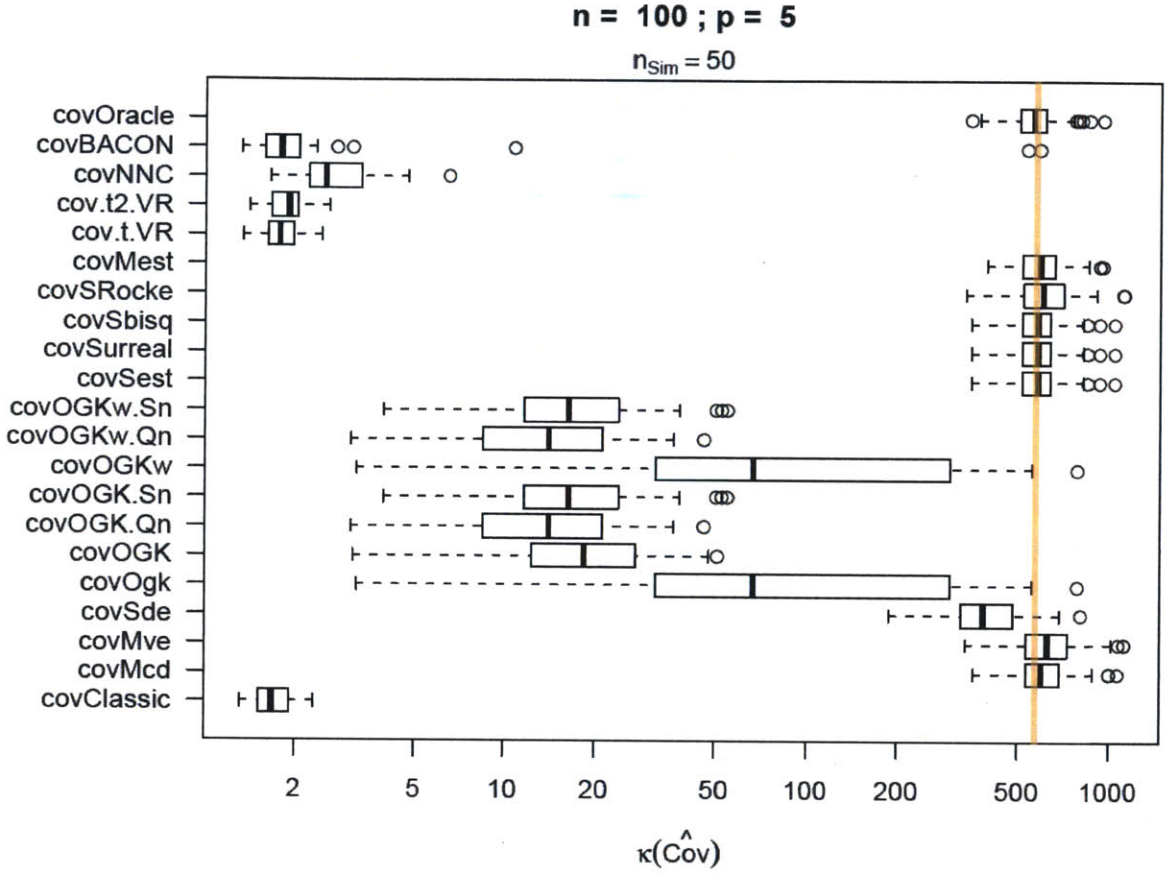


Figure 2.5.4 shows a comparison of computation times for D-Vine (LTS) and MCD. It is clear that, with large n, D-Vine (LTS) runs much faster than MCD. Also, the difference grows much more significantly as n becomes larger. Finally, note that our proposed robust covariance estimator may not necessarily be affine equivariant.

Figure 2.5.3 - Benchmark Results of Our Estimators

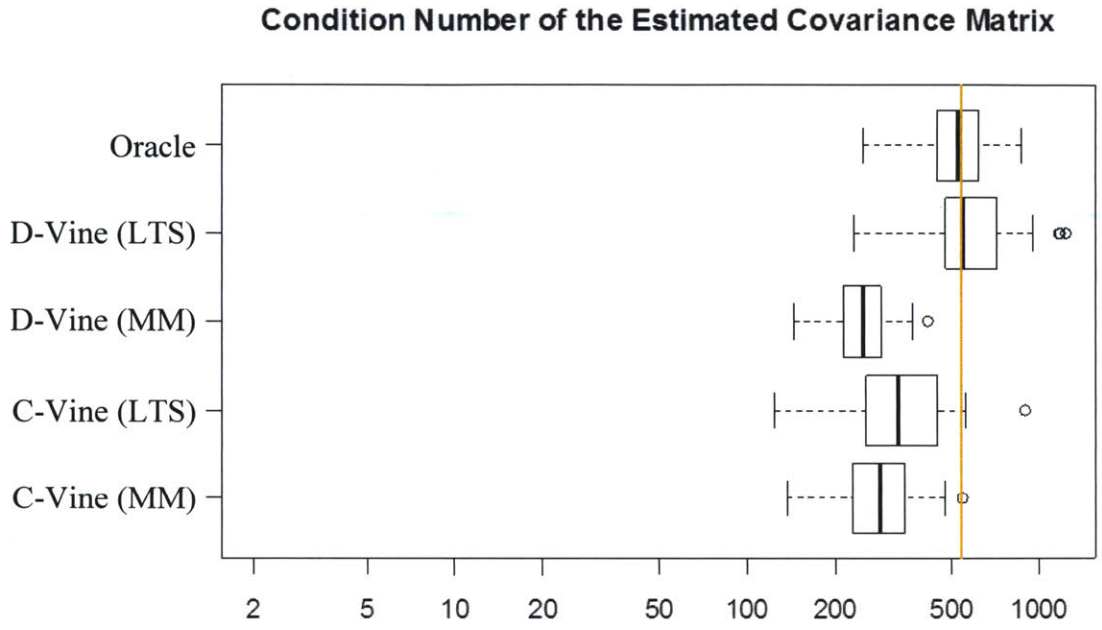
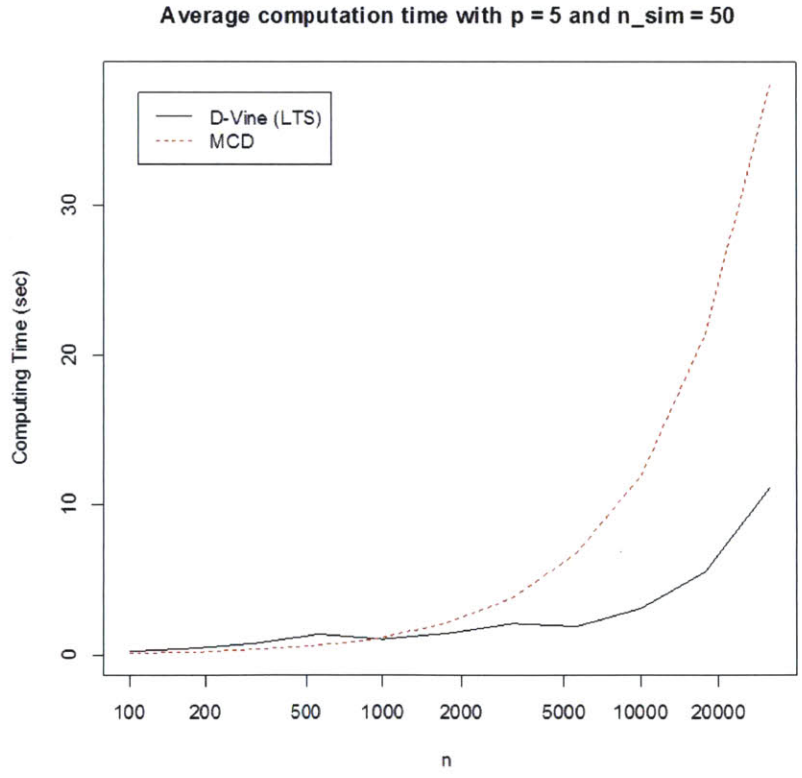


Figure 2.5.4 - Comparison of Computation Time between D-vine (LTS) and MCD



2.6. Financial Application

Covariance/correlation estimations are of significant importance in the field of finance. Many areas, including but not limited to asset allocation and active portfolio management, hedging instrument selection, and credit derivative pricing/modeling, require covariance/correlation in the modeling phase, and they are subject to estimation errors because it is well known that most financial data series are not normally distributed and have skewed tails. As we will demonstrate in an example of asset allocation, our proposed robust correlation estimator is superior to many existing methods.

The Capital Asset Pricing Model (CAPM) is a classical asset allocation method proposed by Markowitz in 1952, and this model and its enhanced versions still remain a very prominent approach to asset allocation and active portfolio management (Garcia-Alvarez and Luger, 2011). The CAPM model minimizes the variance of the portfolio with a target return, and therefore the expected returns and the covariance matrix are the inputs for the model. In practice, both the expected return and the covariance matrix have to be estimated from the data, and therefore, the model is subject to estimation errors. It is well-known that the naive implementation of the CAPM model, with the sample mean and sample covariance matrix as inputs, performs poorly out of sample (Garcia-Alvarez and Luger, 2011). Furthermore, according to a study by DeMiguel, Garlappi and Uppal (2009), the CAPM model, together with many enhanced estimation methods for the mean and covariance matrix, does not perform better than the simple equal-weight portfolio. The equal-weight portfolio simply invests equally in all N stocks under consideration, so it is not subject to estimation errors. This chapter will demonstrate that the mean-variance portfolio selection using our proposed robust covariance estimator outperforms many popular allocation methods in the literature, including the equal-weight portfolio.

20 stocks have been selected from S&P 500 index based on their GICS sectors, market capitalizations and availability of data. The ticker symbols are MCD, DIS, WMT, PG, XOM, CVX, WFC, JPM, JNJ, PFE, GE, UTX, AAPL, MSFT, DD, DOW, T, VZ, SO, D,

and the order of the stock is based on their GICS sectors. Historical stock price data were downloaded from Yahoo! Finance using the close price adjusted for dividends and splits.

We rebalance portfolio weights weekly for the last 15 years, which include both the recent and 1998 financial crisis with extreme stock returns. For each weekly rebalance, we use the past 200 weeks of stock return observations for the estimation process. We will consider and compare returns of the following 6 portfolio allocation rules: D-vine, equal-weight, OGK, MCD, CAPM and S&P 500. Obviously, both equal-weight and S&P 500 portfolio allocation rules do not require any model or estimation. The remaining models use the mean-variance optimization model with different estimators for the location (mean) vector and the dispersion (covariance) matrix.

Estimation methods for each allocation rule:

- D-vine
    - The location vector is estimated using the sample median vector;
    - The dispersion matrix is estimated using D-vine as the partial correlation vine structure and LTS with 50% breakdown as the robust partial correlation estimator.
    - The order of first level tree for the D-vine is the same as the order of the stock as specified above.
- OGK
    - The location vector and the dispersion matrix are estimated using OGK (Maronna and Zamar, 2002) with median and median absolute deviation (MAD) for the univariate robust location and scale estimates.
- MCD
    - The location vector and the dispersion matrix are estimated using the Fast MCD algorithm (Rousseeuw and Driessen, 1999) with 50% breakdown.
- CAPM
    - The location vector is estimated using the classical sample mean;
    - The dispersion matrix is estimated using the classical sample covariance.

The target return is the return of the equal-weight portfolio. However, due to the differences in estimation methods for the location vector of the return, the estimated target return under different methods may not be identical. In addition, short selling is not allowed.

For the mean-variance optimal allocation methods, the following optimization problem is solved with different estimated location vectors and dispersion matrices.

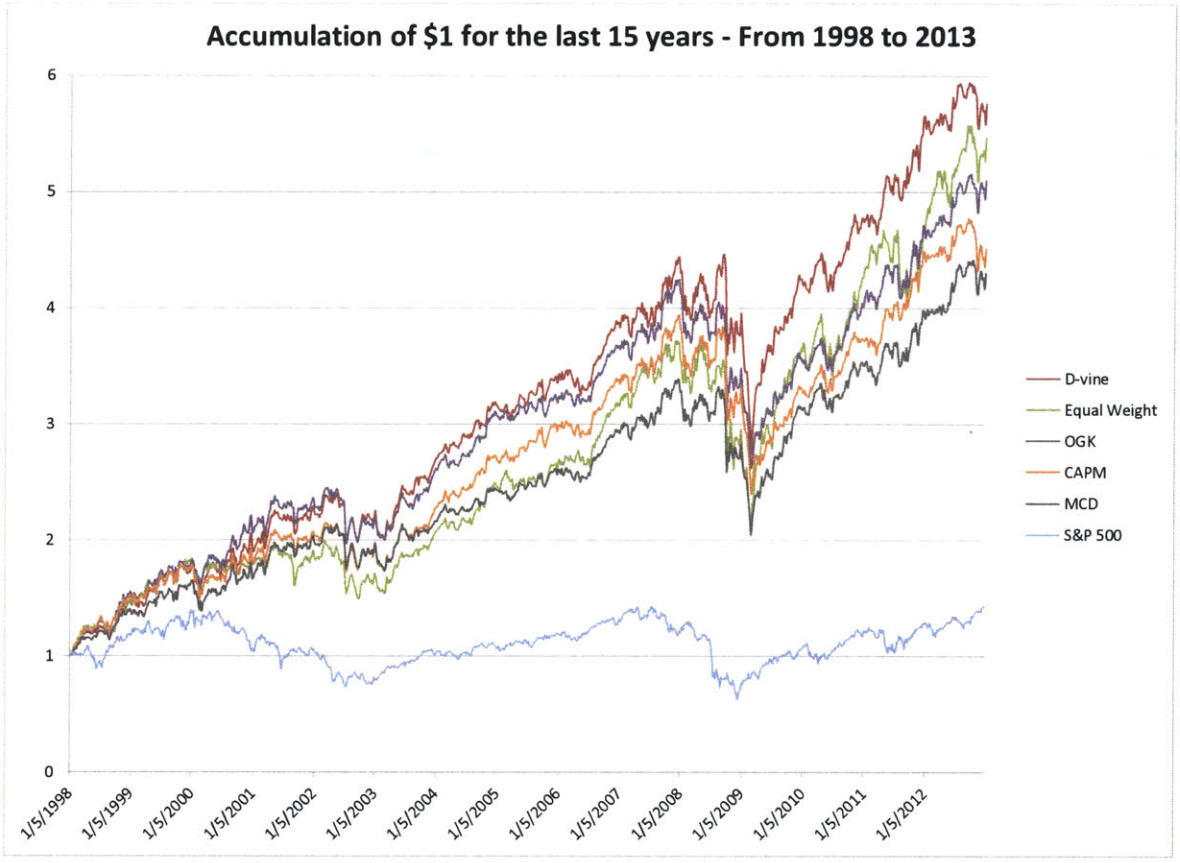
$$\begin{aligned} &\text{Minimize} && w'\hat{\Sigma}w \\ &\text{Subject to:} && w'\hat{\mu} = \mu_{target} \\ & && w'e = 1 \\ & && w \geq 0 \end{aligned}$$

where w is the weight vector, $\hat{\mu}$ is the estimated location (mean) vector, $\hat{\Sigma}$ is the estimated dispersion (covariance) matrix, μ_{target} is the target return location for the portfolio, and e is the vector of all ones.

Finally, no transaction cost is included. Transaction costs and limits on stock turn-overs will be considered in future research.

Figure 2.6.1 shows the accumulation of $1 for the last 15 years under the six different asset allocation methods.

Figure 2.6.1 - Comparison of Portfolio Performance between Each Asset Allocation Method



It is obvious that throughout the trajectory, the allocated portfolio using the D-Vine estimator stays as a top performer, and it does well recovering from both financial crisis of 1998 and 2008.

Table 2.6.1 - Statistics of the Realized Weekly Log-Returns for Each Asset Allocation Method

	D-Vine	Equal Weight	OGK	MCD	CAPM	S&P 500
Mean	**0.2242%**	0.2175%	0.2086%	0.1868%	0.1930%	0.0450%
s.d.	2.0580%	2.4299%	1.9888%	2.0077%	**1.9829%**	2.6724%
IR	**5.5350%**	4.8743%	5.1301%	4.4591%	4.6318%	N/A

where the information ratio $IR = \frac{E[R_p - R_b]}{\sqrt{Var(R_p - R_b)}}$, and the benchmark return, R_b, is the S&P 500 return.

From Table 2.6.1, the allocation method using the Selected Vine has the highest IR for the realized returns, and therefore, it is considered the best active asset allocation method under the IR criterion. In particular, its realized returns actually have higher mean and higher IR compared to the equal-weight portfolio, and therefore, our proposed method delivers better performance than the equal-weight allocation rule.

The asset allocation application has demonstrated that our proposed robust covariance estimation can deliver better results than many existing methods. Also, by estimating the location vector and the dispersion matrix robustly, one can easily identify outliers using the robust version of the Mahalanobis distance, and any data point with a Mahalanobis distance above a certain threshold value would be considered an extreme observation. This is very useful in risk management when modeling the different correlation/dependence structures under normal and crisis situations.

2.7. Discussion

This chapter introduced a new robust covariance estimation method for multivariate data. This method combines the regular vine structure with the robust partial correlation estimation technique. It guarantees the positive definiteness of the correlation/covariance matrix, which is an important property for robust covariance estimators using pair-wise covariance estimates. The breakdown point is also preserved in the estimation process, and this allows us to build estimators with high breakdown point. Then, the Barrow Wheel benchmark shows that this new approach effectively captures the good portion of the data while improving the computing time for large datasets. Finally, we demonstrate the benefit of using our proposed estimation method in the application of active asset allocation, and it delivers superior results when compared to many existing methods.

Chapter 3

Selecting Robust Dependence Model for High-Dimensional Covariance Matrix

In this chapter, we propose an algorithm for selecting regular vines using information from the dataset, and then this selection methodology is applied in two applications.

3.1. Introduction

In Chapter 2, we introduced the regular vine dependence structure for robust modeling of high-dimensional covariance matrices. Two special cases of the regular vine, namely C-Vine and D-Vine, were defined and used throughout the previous chapter, and the D-Vine structure was applied in the asset allocation example. However, there are many more different types of regular vines that are neither C-Vine nor D-Vine, and in many applications, restricting our selection to just C-Vine and D-Vine is not appropriate. In these situations, a vine selection technique is of critical importance, and without the help from an expert, selecting an appropriate vine structure using the information alone from the dataset is the only choice.

3.2. Literature Review

There are several vine selection techniques proposed in the existing literature, and we cover three of the most prominent ones.

Dissmann J, Brechmann E C, Czado C, Kurowicka D (2012) proposed a top-down sequential method to select a vine structure based on Kendall's tau. This method constructs the vine by building the trees one level at a time, and it works as follows:

1) Compute the Kendall's tau for all possible variable pairs
2) Select the spanning tree that maximizes the sum of absolute value of the Kendall's taus, and the resulting tree is the first level tree in the vine
3) For each edge in the above spanning tree, select a copula and estimate the corresponding parameters, and then transform the conditional distributions using the fitted copula
4) For tree level $i = 2, \ldots, p-1$
    a. Compute the Kendall's tau for all conditional variable pairs that satisfy the proximity condition
    b. Select the spanning tree that maximizes the sum of absolute value of the Kendall's taus, and the resulting tree is the i^{th} level tree in the vine
    c. For each edge in the above spanning tree, select a conditional copula and estimate the corresponding parameters, and then transform the conditional distributions using the fitted copula

This method provides a systematic way to find the vine structure, and the use of Kendall's tau provides some level of robustness. However, the need to choose a copula and estimate the parameters makes this method vulnerable to estimation errors, and the estimation error gets worse because later level trees are dependent on the previous copula choices and estimations.

Kurowicka and Joe (2011), Chapter 11 "Optimal Truncation of Vines", proposed a bottom-up approach to select a vine structure based on partial correlations. The algorithm is rather technical, but the concept is as follows:

1) Compute the normalized inverse matrix for the correlation matrix, and choose the smallest absolute partial correlation. The corresponding edge is the edge for the last level tree T_{p-1}

2) Use the algorithm in the paper to build the trees backwards, from $p-2$ to 1

This method again provides a systematic way to find the vine structure, and it does not require any choice or estimation of the copula structure. However, it uses the classical partial correlation without any consideration of robustness, and the use of the covariance matrix in the first step is not applicable when the goal is to estimate the covariance matrix.

Kurowicka, Cooke and Callies (2006) proposed a heuristic search that adds a variable to the vine one at a time instead of working level by level. The heuristic works as follows:

1) Specify an ordering for the variables
2) Start with the vine consisting the first 2 ordered variables
3) For variables from 3 to p (where p is the number of variables), extend the existing vine to include the new variable such that the new vine has a minimum entropy function of $\ln(1-\rho^2_{C_e;D_e})$
4) Store the final vine obtained when all variables are included
5) Repeat from step 1 again for several times, and select the optimal vine that minimizes the entropy function among all those stored

Obviously it is not possible to search through all vines, so randomized sampling is used in practice. Also, this method uses the classical partial correlation without any consideration of robustness, so it is most useful when the underlying data is elliptically distributed without many outliers.

This chapter will introduce a new robust vine selection technique that minimizes the entropy function for the central piece of the data. This selection technique is robust to outliers, and it does not need many distributional assumptions or any distribution/copula estimations in the process.

3.3. Regular Vine Correlation Computation Algorithm

After having estimated, possibly robustly, all the partial correlations on a regular vine, we need to construct the correlation matrix corresponding to the partial correlation vine specifications. Bedford & Cooke (2002) proved the following theorem:

Theorem 3.3.1 (Bedford and Cooke, 2002). For any regular vine on p elements, there is a one-to-one correspondence between the set of $p \times p$ positive definite correlation matrices and the set of partial correlation specifications for the vine.

The Bedford & Cooke (2002) paper stated that "unconditional correlations can easily be calculated inductively," and it also demonstrated this statement in a real computational example. However, the paper did not provide a systematic algorithm for computing unconditional correlations for regular vines in general, and the example it gave was for a D-vine, a very special case of the regular vine family.

In this section, we will provide an efficient, i.e. polynomial time, computational algorithm for deriving the unconditional correlation matrix from any regular partial correlation vine. We will first introduce the matrix representation for regular vines. Then, with its help, we can identify all the sub-vines and derive all unconditional correlations sequentially.

The matrix representation of regular vines is a much more convenient way of representing a regular vine than drawing trees for all levels. The matrix stores all the constraint sets of the regular vine.

Definition 3.3.1 (Constraint set, Conditioning set and Conditioned set).

1. For $e \in \mathcal{E}_i$, $i \leq p-1$, the **constraint set** associated with e is the complete union U_e^* of e, i.e. the subset of $\{1, \ldots, p\}$ reachable from e by the membership relation.
2. For $i = 1, \ldots, p-1$, $e \in \mathcal{E}_i$, if $e = \{j, k\}$, then the **conditioning set** associated with e is

$$D_e = U_j^* \cap U_k^*$$

and the **conditioned set** associated with e is

$$\{C_{e,j}, C_{e,k}\} = \{U_j^* \backslash D_e, U_k^* \backslash D_e\}$$

3. The constraint set for the regular vine $\mathcal{V}$ is a set:

$$\mathcal{CV} = \{(\{C_{e,j}, C_{e,k}\}, D_e) | e \in \mathcal{E}_i, e = \{j,k\}, i = 1, \dots, p-1\}$$

Obviously, for $e \in \mathcal{E}_1$, the conditioning set is empty. Also, the order of an edge is the cardinality of its conditioning set. For $e \in \mathcal{E}_i$, $i \leq p-1$, $e = \{j,k\}$, we have $U_e^* = U_j^* \cup U_k^*$.

Morales-Napoles (2008) and Diβmann, Brechmann, Czado and Kurowicka (2012) uses a lower triangular matrix to store a regular vine structure. Specifically, the matrix stores the constraint sets of a regular vine in columns of an n- dimensional lower triangular matrix. We will first define the concept of matrix constraint set, which stores all the constraint sets in the matrix. Then we will give a concrete example of the construction of such matrix.

Definition 3.3.2 (matrix constraint set). Let $M = (m_{i,j})_{i,j=1,\dots,p}$ be a lower triangular matrix. The i-th constraint set for M is

$$\mathcal{C}_M(i) = \{(\{m_{i,i}, m_{k,i}\}, D) | k = i+1, \dots, p, D = \{m_{k+1,i}, \dots, m_{p,i}\}\}$$

for $i = 1, \dots, p-1$. If $k = p$, we set $D = \emptyset$. The constraint set for matrix M is the union $\mathcal{CM} = \cup_{i=1}^{p-1} \mathcal{C}_M(i)$. For the elements of the constraint set $(\{m_{i,i}, m_{k,i}\}, D) \in \mathcal{CM}$, we call $\{m_{i,i}, m_{k,i}\}$ the conditioned set and D the conditioning set.

The idea is that for every regular vine $\mathcal{V}$, there exists a lower triangular matrix M such that $\mathcal{CV} = \mathcal{CM}$. Therefore, we can encode all information of the vine $\mathcal{V}$ into M.

Definition 3.3.3 (Regular Vine Matrix). A lower triangular matrix $M = (m_{i,j})_{i,j=1,\dots,p}$ is called a **regular vine matrix** if for $i = 1, \dots, p-1$ and for $k = i+1, \dots, p-1$ there is a j in $i+1, \dots, p-1$ with

$$(m_{k,i}, \{m_{k+1,i}, \dots, m_{p,i}\}) \in B_M(j) \text{ or } \in \tilde{B}_M(j)$$

where

$$B_M(j) := \{(m_{j,j}, D) | k = j+1, \dots, p; D = \{m_{k,j}, \dots, m_{p,j}\}\}$$

$$\tilde{B}_M(j) := \{(m_{k,j}, D) | k = j+1, \dots, p; D = \{m_{j,j}\} \cup \{m_{k+1,j}, \dots, m_{p,j}\}\}$$

The condition in Definition 3.3 is to ensure the proximity condition in the definition of a regular vine.

Proposition 3.3.1 (Properties of the Regular Vine Matrix). Let a lower triangular matrix $M = (m_{i,j})_{i,j=1,\dots,p}$ be a regular vine matrix, then

1. $\{m_{i,i}, \dots, m_{p,i}\} \subset \{m_{j,j}, \dots, m_{p,j}\}$ for $1 \leq j < i \leq p$;
2. $m_{i,i} \notin \{m_{i+1,i+1}, \dots, m_{p,i+1}\}$ for $i = 1, \dots, p-1$;
3. All elements in a column are different;
4. Deleting the first row and column from M gives an $(p-1)$-dimensional regular vine matrix.

Proposition 3.3.1.1 states that every column contains all the entries in the columns to the right. Proposition 3.3.1.2 states that diagonal elements are all distinct, and this property is further used in Definition 3.3.4. Proposition 3.3.1.4 states that all sub-vines can be obtained by sequentially removing rows and columns.

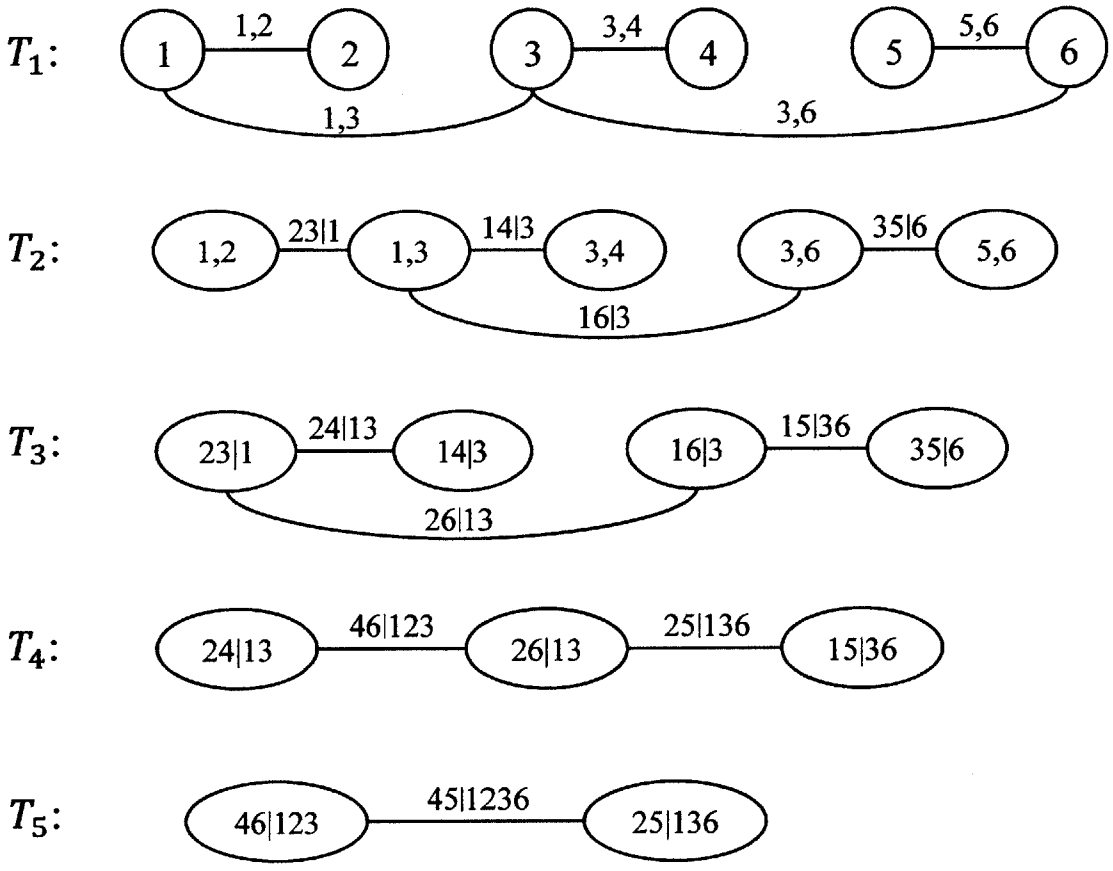
Definition 3.3.4 (Natural Order). For any regular vine matrix $M = (m_{i,j})_{i,j=1,\dots,p}$, the natural order of M, or $NO(M)$, is

$$NO(M) = (m_{1,1}, \dots, m_{p,p})$$

By Proposition 3.3.1.2, $NO(M)$ contains n distinct numbers from 1 to p.

We will demonstrate the concepts in the following example.

Figure 3.3.1 - Sample Regular Vine $\mathcal{V}^*$



One possible matrix representation for $\mathcal{V}^*$ in figure 3.3.1 is M^* shown below. There is another way of constructing the matrix by starting with $M^*_{1,1} = 5$.

$$M^* = \begin{bmatrix} 4 & & & & & \\ 5 & 5 & & & & \\ 6 & 2 & 2 & & & \\ 2 & 1 & 6 & 6 & & \\ 1 & 3 & 3 & 1 & 1 & \\ 3 & 6 & 1 & 3 & 3 & 3 \end{bmatrix}$$

For example, for $i = 1, k = 3$, so $m_{i,i} = 4$, $m_{k,i} = 6$, and $D = \{2,1,3\}$. This corresponds to the constraint set $(\{4,6\},\{2,1,3\})$, which is the first edge in T_4. One can check that the matrix constraint sets of M^* match exactly with the constraint sets of the vine $\mathcal{V}^*$.

The natural order of M^*, or $NO(M^*)$, is $(4,5,2,6,1,3)$.

Diβmann (2010) has proven that there is an equivalent regular vine matrix, not necessarily unique, encoding all the constraint sets for every regular vine. Note that there are different matrix representations for the same regular vine.

Now, we will propose an efficient, $O(p^{4.373})$ runtime, algorithm for computing all unconditional correlations for any given regular partial correlation vine.

For any regular partial correlation vine $\mathcal{V}$, let M be a matrix representation of $\mathcal{V}$. For simplicity, from now on we can assume WLOG that $NO(M) = (p, p-1, \ldots, 1)$ because we can reorder all the variables.

Theorem 3.3.2 (Anderson, 1958). With $a_1, \ldots, a_n$ being all distinct,

$$\rho_{a_1,a_n} = r_1'(a_1,\ldots,a_n)R_2(a_1,\ldots,a_n)^{-1}r_3(a_1,\ldots,a_n) + \rho_{a_1,a_n;a_2,\ldots,a_{n-1}}D_{a_1,\ldots,a_n} \qquad (3.3.1)$$

where

- $r_1'(a_1,\ldots,a_n) = (\rho_{a_1,a_2},\cdots,\rho_{a_1,a_{n-1}})$
- $r_3'(a_1,\ldots,a_n) = (\rho_{a_n,a_2},\cdots,\rho_{a_n,a_{n-1}})$
- $R_2(a_1,\ldots,a_n) = \begin{pmatrix} 1 & \rho_{a_2,a_3} & \cdots & \rho_{a_2,a_{n-1}} \\ \rho_{a_3,a_2} & 1 & & \rho_{a_3,a_{n-1}} \\ \vdots & & \ddots & \vdots \\ \rho_{a_{n-1},a_2} & \rho_{a_{n-1},a_3} & \cdots & 1 \end{pmatrix}$
- $D^2_{a_1,\ldots,a_n} = \left(1 - r_1'(a_1,\ldots,a_n)R_2(a_1,\ldots,a_n)^{-1}r_1(a_1,\ldots,a_n)\right) \cdot \left(1 - r_3'(a_1,\ldots,a_n)R_2(a_1,\ldots,a_n)^{-1}r_3(a_1,\ldots,a_n)\right)$

With Theorem 3.3.2, we can compute all the unconditional correlations sequentially.

Algorithm 3.3.1 (Correlation matrix from regular partial correlation vine).

for $i = 2, \dots, p$ **do**

  for $j = p, \dots, p - i + 2$ **do**

    set $n = p - j + 2$

    set $a_1 = i$

    set $a_n = m_{j,p-i+1}$

    for $k = 2, \dots, n - 1$ **do**

      set $a_k = m_{j+k-1,p-i+1}$

    end for

    use Theorem 3.2 to compute ρ_{a_1,a_n}, i.e. $\rho_{i,m_{j,p-i+1}}$

  end for

end for

This algorithm simply works from the second right-most column and derive correlations from partial correlations inductively. The runtime of this algorithm is $O(p^{4.373})$ because matrix inversion for $p \times p$ matrix takes $O(p^{2.373})$ runtime (Williams, 2011).

In Appendix 3.A, we provide a detailed demonstration by applying Algorithm 3.3.1 on the vine shown in Figure 3.3.1.

3.4. Robust Correlation Vine Structure Selection

In the previous chapter, we proposed a new robust covariance estimator using any pre-specified partial correlation vine structure. In this section, we will discuss how to select the best partial correlation vine structure for a given data set.

In the previous chapter, we have demonstrated estimating robust covariance under both C-Vine and D-Vine. It can be checked that, with $p \leq 4$, every regular vine is either a C-Vine or a D-Vine. However, with $p > 4$, there exists regular vine that is neither a C-Vine or a D-Vine. The regular vine in Figure 3.3.1 is such an example with $p = 6$.

Definition 3.4.1 (Multivariate Elliptical Distribution). $\boldsymbol{X}$ has a multivariate elliptical distribution with location vector $\boldsymbol{\mu}$ and dispersion matrix Σ if its characteristic function can be expressed as

$$\varphi_{\boldsymbol{X}}(\boldsymbol{t}) = E\left[e^{i\boldsymbol{t}^T\boldsymbol{X}}\right] = e^{i\boldsymbol{t}^T\boldsymbol{\mu}}\Psi(\boldsymbol{t}^T\Sigma\boldsymbol{t})$$

If $\boldsymbol{X}$ has a probability density function $f_{\boldsymbol{X}}$, it must be in the form of

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{c}{\sqrt{|\Sigma|}} g\left((\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

In traditional robust statistical analysis, the central data is assumed to be normally distributed, but we want to broaden such assumption to be the entire elliptical family, which includes the multivariate normal distribution, but also heavy tail distributions like the multivariate t-distribution.

Theorem 3.4.1 (Entropy of Elliptical Distribution). Let $\boldsymbol{X}$ be a multivariate elliptical distribution with location vector $\boldsymbol{\mu}$ and dispersion matrix Σ. Let $Z = \Sigma^{-\frac{1}{2}}(X-\mu)$ be the standardized version of X, whose density function does not depend on $\boldsymbol{\mu}$ and Σ. Let $H(\cdot)$ be the entropy function, then

$$H(\boldsymbol{X}) = \frac{1}{2}\ln(|\Sigma|) + H(\boldsymbol{Z})$$

Note that $H(\boldsymbol{Z})$ does not depend on $\boldsymbol{\mu}$ and Σ.

Under Theorem 3.4.1, when we minimize the entropy of X by varying only $\boldsymbol{\mu}$ and Σ, we can actually minimize $\ln(|\Sigma|)$ or $|\Sigma|$ since $\ln(\cdot)$ is a strictly increasing function. This provides us with a theoretical foundation for selecting the optimal vine structure in terms of entropy. Therefore, when searching for the vine that produces the minimum entropy, we

simply have to search for the one that produces the minimum determinant of the resulting covariance matrix.

Formulation 3.4.1 (Optimal Vine Selection Formulations). We will use the following formulation for selection the optimal vine:

$$\begin{aligned} &\text{Minimize} && \left|\hat{\Sigma}(\mathcal{V}, X)\right| \\ &\text{Subject to} && \mathcal{V} \text{ is a regular vine} \end{aligned}$$

where X is the data, and $\hat{\Sigma}(\mathcal{V}, X)$ is the function that produces the estimated covariance matrix using vine $\mathcal{V}$ from data X.

In Chapter 2, the proposed robust covariance matrix estimator follows a three-step process, which is stated in Formulation 3.4.2 below.

Formulation 3.4.2. Let $X = (X_1, \ldots, X_p)$ be the data:

1. Estimate $\widehat{\sigma_1}, \ldots, \widehat{\sigma_p}$ without specifying any vine structure.
2. Given a specified vine structure $\mathcal{V}$, estimate the correlation matrix $\hat{\mathrm{P}}(\mathcal{V}, X)$.
3. Produce the covariance matrix estimate $\hat{\Sigma}(\mathcal{V}, X) = \hat{S} \times \hat{\mathrm{P}}(\mathcal{V}, X) \times \hat{S}$, where $\hat{S} = diag(\widehat{\sigma_1}, \ldots, \widehat{\sigma_p})$.

Proposition 3.4.1. Under Formulation 3.4.2, and assuming that none of the marginal covariance estimates is zero (which is reasonable in practice), then the following optimal vine selection formulations are equivalent:

1.

$$\begin{aligned} &\text{Minimize} && \left|\hat{\Sigma}(\mathcal{V}, X)\right| \\ &\text{Subject to} && \mathcal{V} \text{ is a regular vine} \end{aligned}$$

2.

$$\begin{aligned} &\text{Minimize} && \left|\hat{\mathrm{P}}(\mathcal{V}, X)\right| \\ &\text{Subject to} && \mathcal{V} \text{ is a regular vine} \end{aligned}$$

where $\hat{\mathrm{P}}(\mathcal{V}, X)$ is the function that produces the estimated correlation matrix using vine $\mathcal{V}$ from data X.

Proof:

Since $\hat{\Sigma}(\mathcal{V}, X) = \hat{S} \times \hat{\mathrm{P}}(\mathcal{V}, X) \times \hat{S}$, we have:

$$|\hat{\Sigma}(\mathcal{V}, X)| = |\hat{S}| \times |\hat{\mathrm{P}}(\mathcal{V}, X)| \times |\hat{S}| = |\hat{S}|^2 \times |\hat{\mathrm{P}}(\mathcal{V}, X)| = \left(\prod_{i=1}^{p} \hat{\sigma}_i^2\right) \times |\hat{\mathrm{P}}(\mathcal{V}, X)|$$

$\left(\prod_{i=1}^{p} \hat{\sigma}_i^2\right) > 0$ by the assumption that none of the marginal covariance estimate is zero, and $\left(\prod_{i=1}^{p} \hat{\sigma}_i^2\right)$ is independent from the vine structure $\mathcal{V}$. Therefore, $\left(\prod_{i=1}^{p} \hat{\sigma}_i^2\right)$ is not a function of $\mathcal{V}$, and the first optimization problem can be reformulated to be:

Minimize	$\left(\prod_{i=1}^{p} \hat{\sigma}_i^2\right) \times \lvert\hat{\mathrm{P}}(\mathcal{V}, X)\rvert$
Subject to	$\mathcal{V}$ is a regular vine

This is obviously equivalent to the second optimization problem. ■

Theorem 3.4.2 (Morales-Napoles, 2008). There are $\binom{p}{2} \times (p-2)! \times 2^{\binom{p-2}{2}}$ labelled regular vines in total.

Theorem 3.4.3 (Preservation of Breakdown point (BP)). If each robust partial correlation estimator has BP at least ϵ^*, and we select the optimal vine structure under Formulation 3.4.1, then the resulting correlation matrix also has BP at least ϵ^*.

Proof:

Let $N = \binom{p}{2} \times (p-2)! \times 2^{\binom{p-2}{2}}$, the total number of labelled regular vines, and let $\{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$ be the complete set of labelled regular vines.

Since each robust partial correlation estimator has BP at least ϵ^*, if we apply Theorem 2.4.2, we have for any given regular vine $\mathcal{V}_i$, its associated resulting correlation matrix has BP at least ϵ^*. By Definition 2.4.3 (BP), for each regular vine $\mathcal{V}_i$, there exists a closed and bounded set K_i, which is also bounded away from the boundary (i.e. $[-1,1]^{\binom{p}{2}}$), such that the correlation estimator falls in the set K_i.

Let $K = \bigcup_{i=1}^{N} K_i$, then K is also closed and bounded, and it is also bounded away from the boundary. In addition, all correlation estimators fall in the set K.

The optimal correlation estimator is one of the correlation estimators, and it has to fall in the set K, and therefore, by definition 2.4.3 (BP), the resulting correlation matrix using the optimal vine structure selected under Formulation 3.4.1 also has BP at least ϵ^*. ■

Therefore, to find the optimal vine structure, we can simply solve the optimization problem under Formulation 3.4.1. However, Theorem 3.4.2 has shown that the total number of regular vines grows exponentially with the number of variables in the dataset, so the runtime to find the true global optimal is time-consuming. Hence, we will propose a greedy algorithm to search for an approximation to the true optimal vine.

The following theorem provides the relationship between the determinant of the correlation matrix and the partial correlation vine. We will use this relationship and propose the heuristic for searching the optimal partial correlation vine structure.

Theorem 3.4.4 (Kurowicka and Cooke, 2006). Let D be the determinant of the p-dimensional correlation matrix ($D > 0$). For any partial correlation vine $\mathcal{V}$,

$$D = \prod_{e \in \mathcal{E}(\mathcal{V})} \left(1 - \rho^2_{e_1, e_2; D_e}\right)$$

One way to construct a vine structure is to build the trees in the vine one level at a time. Therefore, at level 1, we have nodes $1,2,\ldots,p$, and we can construct a connected tree, T_1,

based on those nodes. At level 2, we have nodes that are edges of T_1, and we can construct a connected tree, T_2, based on those nodes obeying the proximity condition. Following this process, we can construct trees $T_1, \ldots, T_{p-1}$ sequentially.

Note that in Theorem 3.4.3, we can rewrite the determinant of the correlation matrix as

$$D = \prod_{i=1}^{p-1} \prod_{e \in \mathcal{E}(T_i)} \left(1 - \rho^2_{e_1,e_2;D_e}\right)$$

Therefore, under the sequential construction algorithm, at level i, we are solving the optimization problem (3.4.1)

$$\min_{T_i} \prod_{e \in \mathcal{E}(T_i)} \left(1 - \rho^2_{e_1,e_2;D_e}\right)$$

$$s.t. \quad T_i \text{ is a connected tree}$$

$$\#(e_1 \Delta e_2) = 2, \forall \{e_1, e_2\} \in \mathcal{E}_i \text{ (i.e. proximity condition)}$$

Now, consider the following optimization problem (3.4.2):

$$\max_{T_i} \sum_{e \in \mathcal{E}(T_i)} \left|\rho_{e_1,e_2;D_e}\right|$$

$$s.t. \quad T_i \text{ is a connected tree}$$

$$\#(e_1 \Delta e_2) = 2, \forall \{e_1, e_2\} \in \mathcal{E}_i \text{ (i.e. proximity condition)}$$

Theorem 3.4.5. optimization problems 3.4.1 and 3.4.2 are equivalent, i.e. an optimal tree found in one problem is also an optimal tree in the other problem.

Proof:

First note that, by applying the logarithm operator to the objective function, problem 3.4.1 is equivalent to the following problem (3.4.1'):

$$\min_{T_i} \sum_{e \in \mathcal{E}(T_i)} \ln\left(1 - \rho^2_{e_1,e_2;D_e}\right)$$

$s.t.$ T_i is a connected tree

$\#(e_1 \Delta e_2) = 2, \forall \{e_1, e_2\} \in \mathcal{E}_i$ (i.e. proximity condition)

This is the minimum spanning tree problem with weights $\ln\left(1 - \rho^2_{e_1,e_2;D_e}\right)$ for edge e.

Similarly, problem 3.4.2 is equivalent to the following problem (3.4.2'):

$$\min_{T_i} \sum_{e \in \mathcal{E}(T_i)} -\left|\rho_{e_1,e_2;D_e}\right|$$

$s.t.$ T_i is a connected tree

$\#(e_1 \Delta e_2) = 2, \forall \{e_1, e_2\} \in \mathcal{E}_i$ (i.e. proximity condition)

This is the minimum spanning tree problem with weights $-\left|\rho_{e_1,e_2;D_e}\right|$ for edge e.

Since the minimum spanning tree algorithm depends only on the ranking of the edge weights, and the weights $\ln\left(1 - \rho^2_{e_1,e_2;D_e}\right)$ and $-\left|\rho_{e_1,e_2;D_e}\right|$ have consistent rankings of the edge weights, it is obvious that an optimal tree for 3.4.1' is an optimal tree for 3.4.2', and vice versa.

Therefore, an optimal tree for 3.4.1 is an optimal tree for 3.4.2, and vice versa. ■

For computation efficiency, we assign the following weight function $w(\cdot)$ as

$$w(e) = \begin{cases} \left|\rho_{e_1,e_2;D_e}\right|, if\ \#(e_1 \Delta e_2) = 2 \\ -\infty, o/w \end{cases}$$

If we use the above weight function, and apply the maximum spanning tree algorithm, it will automatically extract the maximum spanning tree that satisfies the proximity condition. By construction the trees level by level, we can find the vine structure that solves the optimization problem (3.4.1).

Dissmann J, Brechmann E C, Czado C, Kurowicka D (2012) proposed a similar algorithm using the maximum spanning tree to search for the optimal vine structure. However, their method requires an additional step of estimating the copula structure at each level, which allows for more estimation error in the process. Our proposed method does not need to use/estimate any distribution/copula structure, and hence it is distribution-free.

Finally, we demonstrate that our proposed optimal vine searching heuristic processes the property of permutation equivariance. This property ensures that regardless how one switches the order of the variables, the resulting covariance/correlation matrix provides the same estimates. Permutation equivariance is a weaker property compared to affine equivariance, which requires the covariance/correlation matrix estimates to be consistent under all affine transformations. Robust estimators such as minimum covariance determinant (MCD) and minimum volume ellipsoid (MVE) process the property of affine equivariance, but they are generally computationally intense and require sampling to begin the process.

Definition 3.4.2 (Affine Equivariance). Let $\hat{\Sigma}$ be a dispersion estimator. Let $X = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$, where each $\boldsymbol{x_i}$ is a p-dimensional observation. We say that $\hat{\Sigma}$ has the property of affine equivariance if

$$\hat{\Sigma}(AX) = A\hat{\Sigma}(X)$$

for all non-singular $p \times p$ matrices A.

Definition 3.4.3 (Permutation Equivariance). Let $\hat{\Sigma}$ be a dispersion estimator. Let $X = \{\boldsymbol{X_1}, \ldots, \boldsymbol{X_p}\}$, where each $\boldsymbol{X_i}$ is a variable. We say that $\hat{\Sigma}$ has the property of permutation equivariance if

$$\hat{\Sigma}\left(\{\boldsymbol{X_{\pi(1)}}, \ldots, \boldsymbol{X_{\pi(p)}}\}\right)_{i,j} = \hat{\Sigma}(X)_{\pi(i),\pi(j)}$$

for all permutations π.

Note: If an estimator $\hat{\Sigma}$ has the property of affine equivariance, then it automatically has the property of permutation equivariance. This is because every permutation has a corresponding permutation matrix. The inverse is obviously not true.

Theorem 3.4.6. Assume that all partial correlation estimates are distinct, which is a reasonable assumption in practice. The correlation estimator under sequential construction of the partial correlation vine is permutation-invariant.

Proof:

Let $X = \{X_1, \dots, X_p\}$ be the original (ordered) data variables.

For any permutation π, then $\tilde{X} = \{X_{\pi(1)}, \dots, X_{\pi(p)}\}$ is the newly ordered data variables, i.e. $\tilde{X}_i = X_{\pi(i)}$.

Let $\mathcal{V} = \{T_1, \dots, T_{p-1}\}$ and $\tilde{\mathcal{V}} = \{\tilde{T}_1, \dots, \tilde{T}_{p-1}\}$ be the vine structures found using the MST heuristic for X and $\tilde{X}$ respectively. Recall Definition 2.3.1, N_j and E_j are nodes and edges for T_j respectively.

We will first show that the vine structures found using the MST heuristic under the two datasets are the same with respect to the variables. Formally, we will prove the following claim:

$$\textbf{At any level } j, \tilde{N}_j = \pi(N_j) \textbf{ and } \tilde{E}_j = \pi(E_j).$$

We will prove this claim by induction on the tree levels.

Base case: $j = 1$, i.e. level 1 tree.

It is obvious $\tilde{N}_1 = \pi(N_1)$ because $\tilde{N}_1 = \{\tilde{X}_i\}_{i=1}^p = \{X_{\pi(i)}\}_{i=1}^p = \pi(\{X_i\}_{i=1}^p) = \pi(N_1)$.

Now, since all partial correlation estimates are distinct by assumption, the maximum spanning tree is uniquely determined. Also, $\rho(\tilde{X}_k, \tilde{X}_l) = \rho(X_{\pi(k)}, X_{\pi(l)})$, and therefore, there is an edge between $(\tilde{X}_k, \tilde{X}_l)$ in $\tilde{T}_1$ iff there is an edge between $(X_{\pi(k)}, X_{\pi(l)})$ in T_1. Hence, $\tilde{E}_1 = \pi(E_1)$.

Inductive hypothesis: assumes true for $j = m - 1$.

Induction step: $j = m$.

By Definition 2.3.1, $\tilde{N}_m = \tilde{E}_{m-1}$, and $N_m = E_{m-1}$, so $\tilde{N}_m = \tilde{E}_{m-1} = \pi(E_{m-1}) = \pi(N_m)$. Using the same argument as in the base case, since all partial correlation estimates are distinct by assumption, the maximum spanning tree is uniquely determined. Therefore, for any two nodes $\{\tilde{a}, \tilde{b}\} \subseteq \tilde{N}_m$ that satisfy the proximity condition, we have $\{\pi(\tilde{a}), \pi(\tilde{b})\} \subseteq N_m$, and it also satisfies the proximity condition. In addition, say $\tilde{a}\Delta\tilde{b} = \{\tilde{X}_k, \tilde{X}_l\}$ and $\tilde{a} \cap \tilde{b} = \{\tilde{X}_{o_1}, \ldots, \tilde{X}_{o_{m-1}}\}$, then

$$\rho_{\tilde{a}\Delta\tilde{b};a\cap\tilde{b}} = \rho_{\tilde{X}_k,\tilde{X}_l;\tilde{X}_{o_1},\ldots,\tilde{X}_{o_{m-1}}} = \rho_{X_{\pi(k)},X_{\pi(l)};X_{\pi(o_1)},\ldots,X_{\pi(o_{m-1})}}$$

Therefore, there is an edge between $\tilde{a}$ and $\tilde{b}$ in $\tilde{T}_m$ iff there is an edge between $\pi(\tilde{a})$ and $\pi(\tilde{b})$. Hence, $\tilde{E}_m = \pi(E_m)$

Therefore, we have shown that the two vine structures under the two datasets are identical with a permutation of the variables. Specifically, if we write down their matrix representations and force their natural orders to be $(p, p-1, \ldots, 1)$, then the two matrices would be identical. Since all the partial correlations are identify after permutation as well, after applying Algorithm 3.3.1, the resulting correlation matrices would be identical after permutation as well, i.e. $\hat{\Sigma}(\{\boldsymbol{X}_{\boldsymbol{\pi}(1)}, \ldots, \boldsymbol{X}_{\boldsymbol{\pi}(\boldsymbol{p})}\})_{i,j} = \hat{\Sigma}(X)_{\pi(i),\pi(j)}$ ■

In Appendix 3.B, we have provided an example showing the importance of the vine selection process.

3.5. Optimality Gap

In the previous section, we proposed the maximum spanning tree heuristic to find the vine structure level by level, and it is an approximation algorithm to finding the vine structure whose covariance/correlation matrix has the minimum determinant. In this section, we investigate the optimality gap between this approximation and the true optimal vine structure.

Recall from the previous section, we are solving the following optimization problem:

$$\begin{aligned} &\text{Minimize} && \sum_{e\in\mathcal{E}(\mathcal{V})} \ln\left(1-\rho^2_{e_1,e_2;D_e}\right) \\ &\text{Subject to} && \mathcal{V} \text{ is a regular vine} \end{aligned}$$

The maximum spanning tree heuristic is an approximation algorithm that uses the greedy method to find the maximum spanning tree at each level tree on the vine that satisfies the proximity condition, and it constructs the vine level by level.

Definition 3.5.1 (Optimality Gap). Let $\mathcal{V}^*$ be the global optimal solution to the optimization problem above, and let $\mathcal{V}^{MST}$ be the solution to the maximum spanning tree heuristic, then the optimality gap between the two, measured in terms of percentage of the difference is:

$$OptGap = \frac{\sum_{e\in\mathcal{E}(\mathcal{V}^*)} \ln\left(1-\rho^2_{e_1,e_2;D_e}\right) - \sum_{e\in\mathcal{E}(\mathcal{V}^{MST})} \ln\left(1-\rho^2_{e_1,e_2;D_e}\right)}{\sum_{e\in\mathcal{E}(\mathcal{V}^*)} \ln\left(1-\rho^2_{e_1,e_2;D_e}\right)}$$

We simulate the optimality gap use the Barrow Wheel benchmark data, and we use the least trimmed square robust regression estimator for the pairwise partial correlation. Unlike regular partial correlations, there is no general relationship between robust partial correlations, and therefore we need to search over all possible vines to fine the true optimal vine. Because the number of regular vines grows exponentially with the dimension of the data, we perform the analyses for $p = 4,5,6$ only.

Table 3.5.1 - Optimality Gap Comparison for Different Data Dimensions

p	Mean	Median	Standard Deviation	Median Absolute Deviation
4	2.51%	1.69%	2.61%	2.49%
5	3.88%	3.65%	2.91%	3.75%
6	5.68%	6.12%	3.11%	3.05%

Figure 3.5.1 - Optimality Gap for $p = 4$

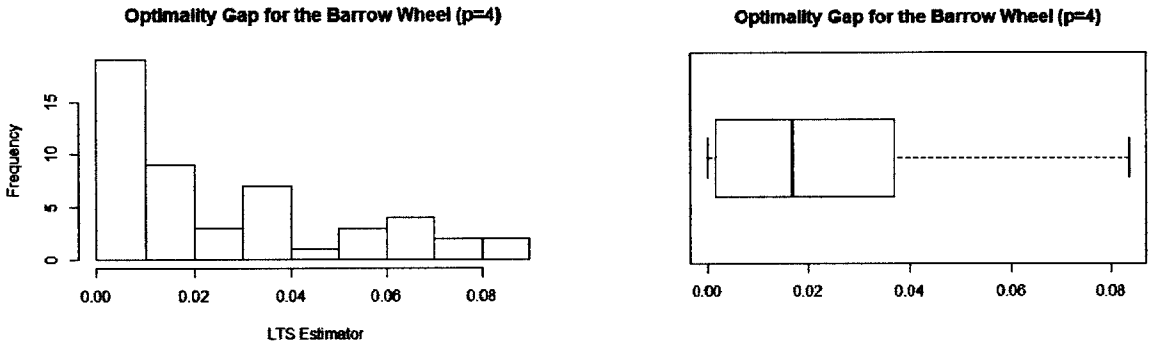


Figure 3.5.2 - Optimality Gap for $p = 5$

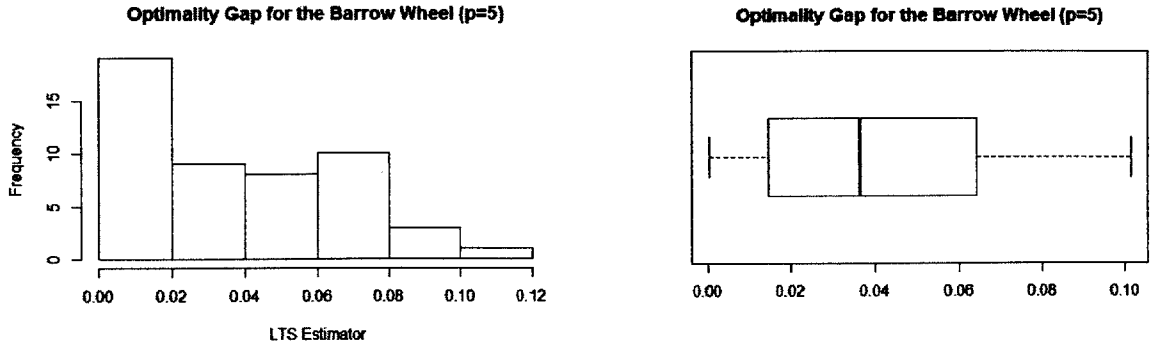


Figure 3.5.3 - Optimality Gap for $p = 6$

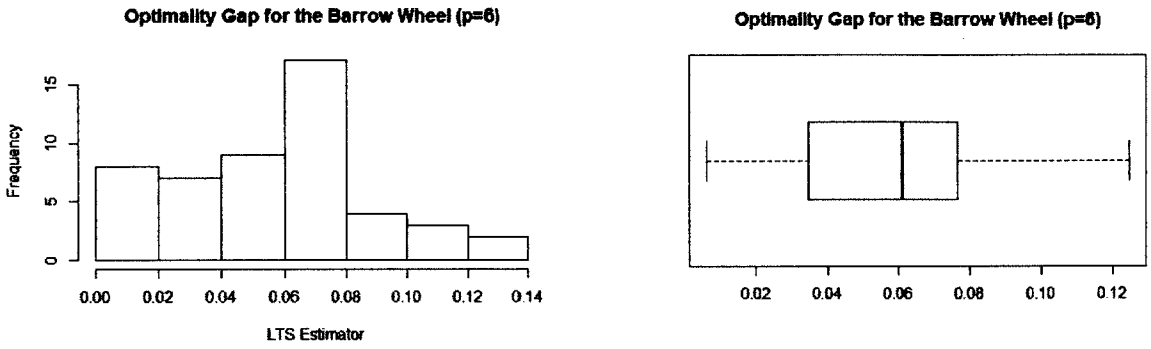


Table 3.5.1 and Figures 3.5.1, 3.5.2 and 3.5.3 show various statistical properties and plots of optimality gaps for different data dimensions, and we conclude that optimality gap is small on average, and the variation of such optimality gap from data set to data set is also small, around 3%.

Finally, we look into the worst case vine for $p = 4$ that has the maximum optimality gap, and we compare the difference between the optimal vine and the vine found by the MST heuristic. Figure 3.5.4 and 3.5.5 show the two vine structures.

Figure 3.5.4 - Optimal Vine Structure

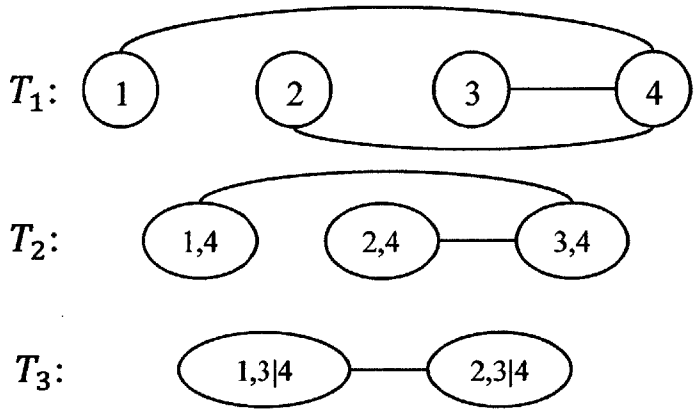
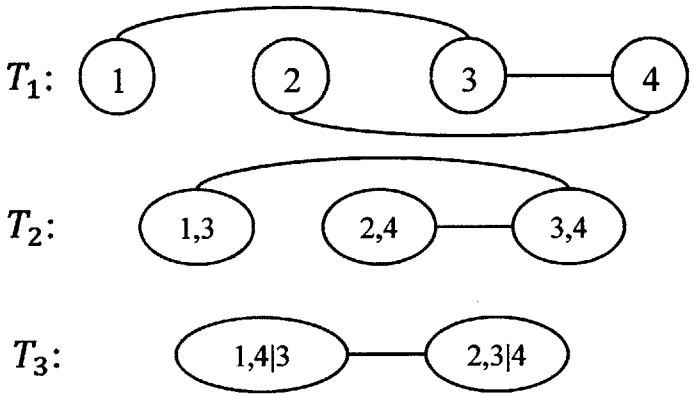


Figure 3.5.5 - Vine Structure Found by the MST Heuristic



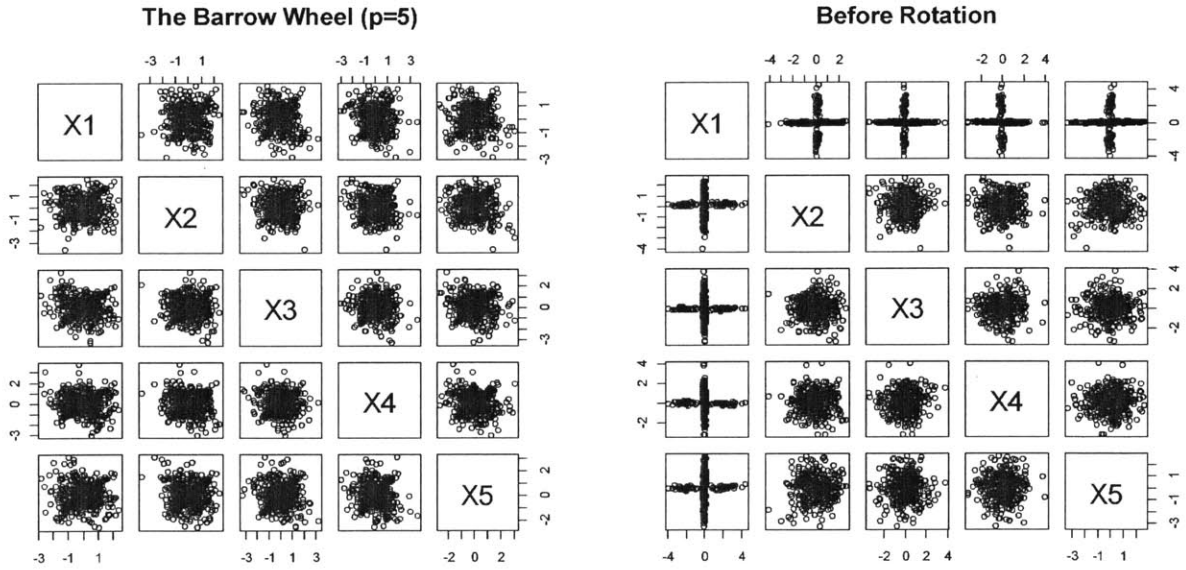
On the first level tree, the optimal vine includes the edge (1,4), while the MST vine includes the edge (1,3). This difference leads to the inclusion of the edge (1,3|4) on the second level tree in the optimal vine and the inclusion of the edge (1,4|3) for the MST vine. All other edges are the same in both vine structures.

Quantitatively, we compute robust partial correlations for all possible pairs for the first level tree: $\rho_{12} = 0.1257$, $\rho_{13} = 0.3675$, $\rho_{14} = 0.3381$, $\rho_{23} = 0.3598$, $\rho_{24} = 0.4668$ and $\rho_{34} = 0.4286$. It is clear that ρ_{13} is larger than ρ_{14}, so (1,3) is selected as an edge over (1,4) when using the maximum spanning tree algorithm. However, on the second level tree, we have the following robust partial correlations: $\rho_{14;3} = 0.1786$ and $\rho_{13;4} = 0.0997$, and this difference is much bigger than the difference between ρ_{13} and ρ_{14}. Therefore, the optimal vine cleverly takes this into consideration while selecting, but the MST vine is just being myopic. However, the optimality gap is just 8.4% for this worst case scenario, so the MST vine selecting heuristic is still an excellent approximation algorithm when searching for the optimal vine structure.

3.6. Benchmark

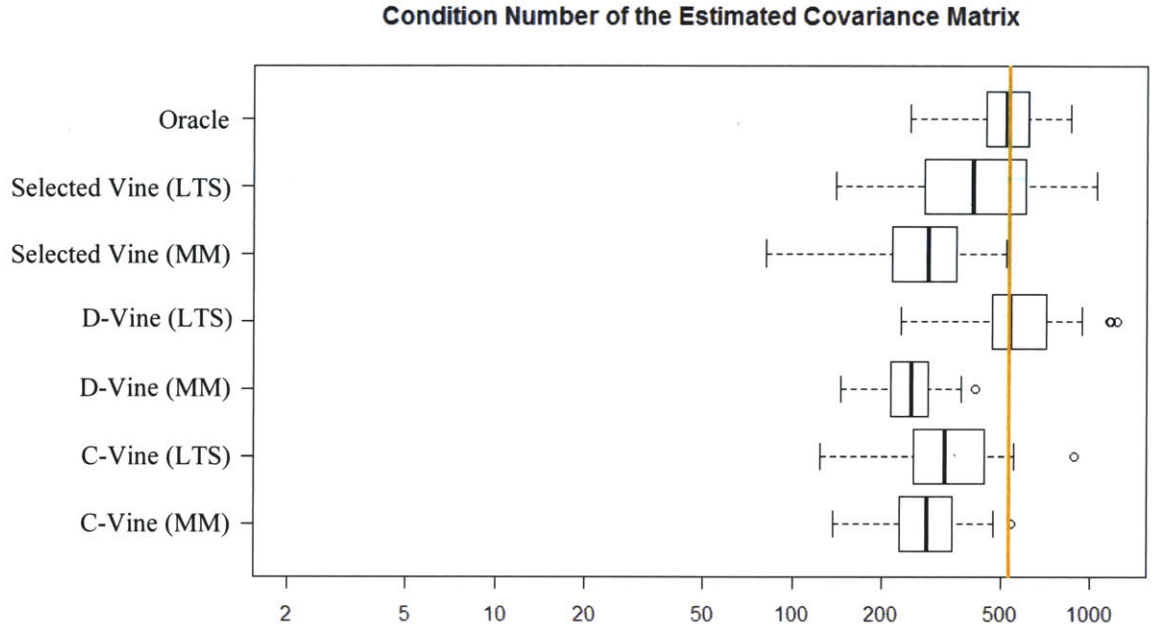
We performed the Barrow Wheel benchmark, which is the same as in Section 2.5, on our proposed MST vine searching heuristic. Recall that the Barrow Wheel distribution is a mixture of a flat normal distribution contaminated with a portion of gross errors concentrated near a one-dimensional subspace, and then this distribution is rotated such that the first variable axis points in the space diagonal direction. Therefore, before the rotation, the first variable component is orthogonal to the remaining variables as shown in Figure 3.6.1.

Figure 3.6.1 - Pairwise Scatter Plot for the Barrow Wheel Distribution (p=5)



The benchmark result is shown in Figure 3.6.2, which is a comparison of the condition number between the selected vine, D-Vine, C-Vine with different robust partial correlation estimators and the oracle. It is clear that the selected vine with LTS robust partial correlation estimator does a good job in terms of condition number.

Figure 3.6.2 - Benchmark Results of Our Estimators



3.7. Financial Application

In this section, we provide two financial applications showing the value of vine selection. The first application is an extension of the asset allocation application in section 2.6, and we include the vine selection when estimating the covariance matrix for portfolio optimization. The second application is an exploration of the underlying dependence structure among several industries and using the industry input-output matrix as a validation of our findings.

For the asset allocation application, the set-up is as follows: 20 stocks have been selected from S&P 500 index based on their GICS sectors, market capitalizations and availability of data. The ticker symbols are MCD, DIS, WMT, PG, XOM, CVX, WFC, JPM, JNJ, PFE, GE, UTX, AAPL, MSFT, DD, DOW, T, VZ, SO, D, and the order of the stock is based on their GICS sectors. Historical stock price data were downloaded from Yahoo! Finance

using the close price adjusted for dividends and splits. We rebalance portfolio weights weekly for the last 15 years, which include both the recent and 1998 financial crisis with extreme stock returns. For each weekly rebalance, we use the past 200 weeks of stock return observations for the estimation process. For simplicity, we will only consider and compare returns of the following 4 portfolio allocation rules: Selected vine, D-vine, equal-weight, CAPM. The models, except the equal-weight, use the mean-variance optimization model with different estimators for the location (mean) vector and the dispersion (covariance) matrix.

Estimation methods for each allocation rule:

- Selected vine
    - The location vector is estimated using the sample median vector;
    - The dispersion matrix is estimated using the MST vine selection heuristic as the partial correlation vine structure and LTS with 50% breakdown as the robust partial correlation estimator.
- D-vine
    - The location vector is estimated using the sample median vector;
    - The dispersion matrix is estimated using D-vine as the partial correlation vine structure and LTS with 50% breakdown as the robust partial correlation estimator.
    - The order of first level tree for the D-vine is the same as the order of the stock as specified above.
- CAPM
    - The location vector is estimated using the classical sample mean;
    - The dispersion matrix is estimated using the classical sample covariance.

The target return is the return of the equal-weight portfolio. However, due to the differences in estimation methods for the location vector of the return, the estimated target return under different methods may not be identical. In addition, short selling is not allowed.

For the mean-variance optimal allocation methods, the following optimization problem is solved with different estimated location vectors and dispersion matrices.

$$\begin{aligned} &\text{Minimize} && w'\hat{\Sigma}w \\ &\text{Subject to:} && w'\hat{\mu} = \mu_{target} \\ & && w'e = 1 \\ & && w \geq 0 \end{aligned}$$

where w is the weight vector, $\hat{\mu}$ is the estimated location (mean) vector, $\hat{\Sigma}$ is the estimated dispersion (covariance) matrix, μ_{target} is the target return location for the portfolio, and e is the vector of all ones.

Finally, no transaction cost is included. Transaction costs and limits on stock turn-overs will be considered in future research.

Figure 3.7.1 shows the accumulation of \$1 for the last 15 years under the four different asset allocation methods.

It is obvious that throughout the trajectory, the allocated portfolio using the Selected Vine estimator stays as a top performer, and it does well recovering from the financial crises of 1998 and 2008.

Figure 3.7.1 - Comparison of Portfolio Performance between Each Asset Allocation Method

Table 3.7.1 - Statistics of the Realized Weekly Log-Returns for Each Asset Allocation Method

	Selected Vine	D-Vine	Equal Weight	CAPM
Mean	**0.2717%**	0.2242%	0.2175%	0.1930%
s.d.	2.5640%	2.0580%	2.4299%	**1.9829%**
IR	**6.2489%**	5.5350%	4.8743%	4.6318%

where the information ratio $IR = \frac{E[R_p - R_b]}{\sqrt{Var(R_p - R_b)}}$, and the benchmark return, R_b, is the S&P 500 return.

From Table 3.7.1, the allocation method using the Selected Vine has the highest IR for the realized returns, and therefore, it is considered the best active asset allocation method under the IR criterion. In particular, its realized returns actually have higher mean and higher IR compared to the equal-weight portfolio, and therefore, our proposed method delivers better performance than the equal-weight allocation rule.

The second application is to use the vine selection technique to uncover the underlying network/dependence structure of many different industries. It is well known that different industries are interrelated, and the recent financial crisis has demonstrated the severity of such linkage that the failure of one part of the economy can seriously affect some other part of the economy. Our proposed vine selection technique is able to find a reasonable dependence structure among different industries, and it can also uncover a deeper relationship between two industries after removing the effect of some other industries.

Six industries were selected from the Global Industry Classification Standard (GICS):

- Consumer Staples
- Financial
- Health Care
- IT
- Telecom
- Utility

For each industry, stock returns of top companies, in terms of market capitalization, were used to uncover the underlying network/dependence structure. Weekly returns from 2008 to 2012, which were downloaded from Yahoo! Finance, were used as the input dataset for the vine selection process. We used the MM regression estimation with high breakdown and efficiency as the robust partial correlation estimator. We used Formulation 3.4.1 to find the true optimal vine structure by enumerating all possible vine structures for 6 variables, so there is no optimality gap.

Figure 3.7.2 - Optimal Vine Structure for the Industry Stock Returns

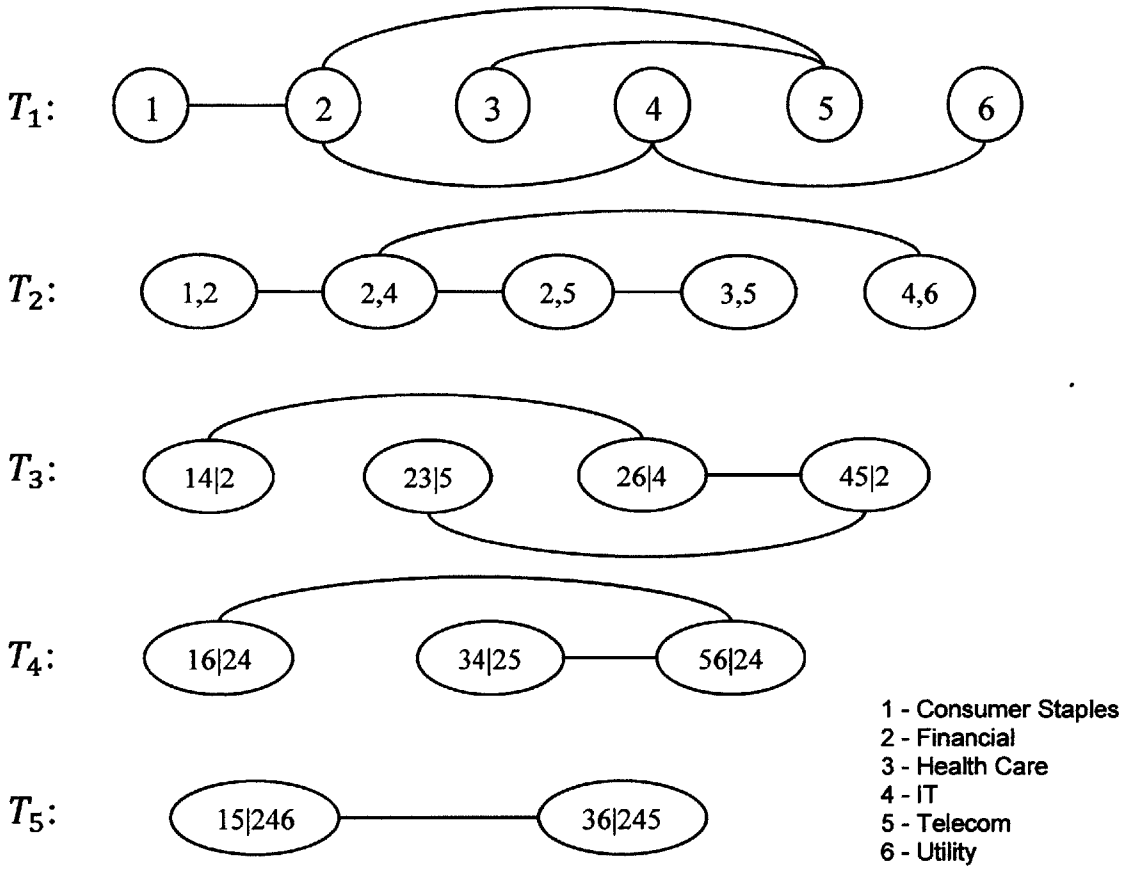


Figure 3.7.2 shows the optimal vine structure selected for the six industries under Formulation 3.4.1. A rough interpretation is that each edge represents strong correlation/partial correlation, either positive or negative, among all possible linkages between the nodes. A more formal approach to interpret the vine, especially the higher level trees, is that this vine provides a foundation to transform the multivariate analysis of these six industries into a pair-wise modeling framework. For example, when managing risk for these industries, one may want to fit a joint distribution/copula structure for the stock returns, but there is very limited range of high-dimensional multivariate copula structure, while there are a handful number of bivariate copula structures. By allowing us to transform this six-dimensional modeling problem into 15 bivariate modeling problem,

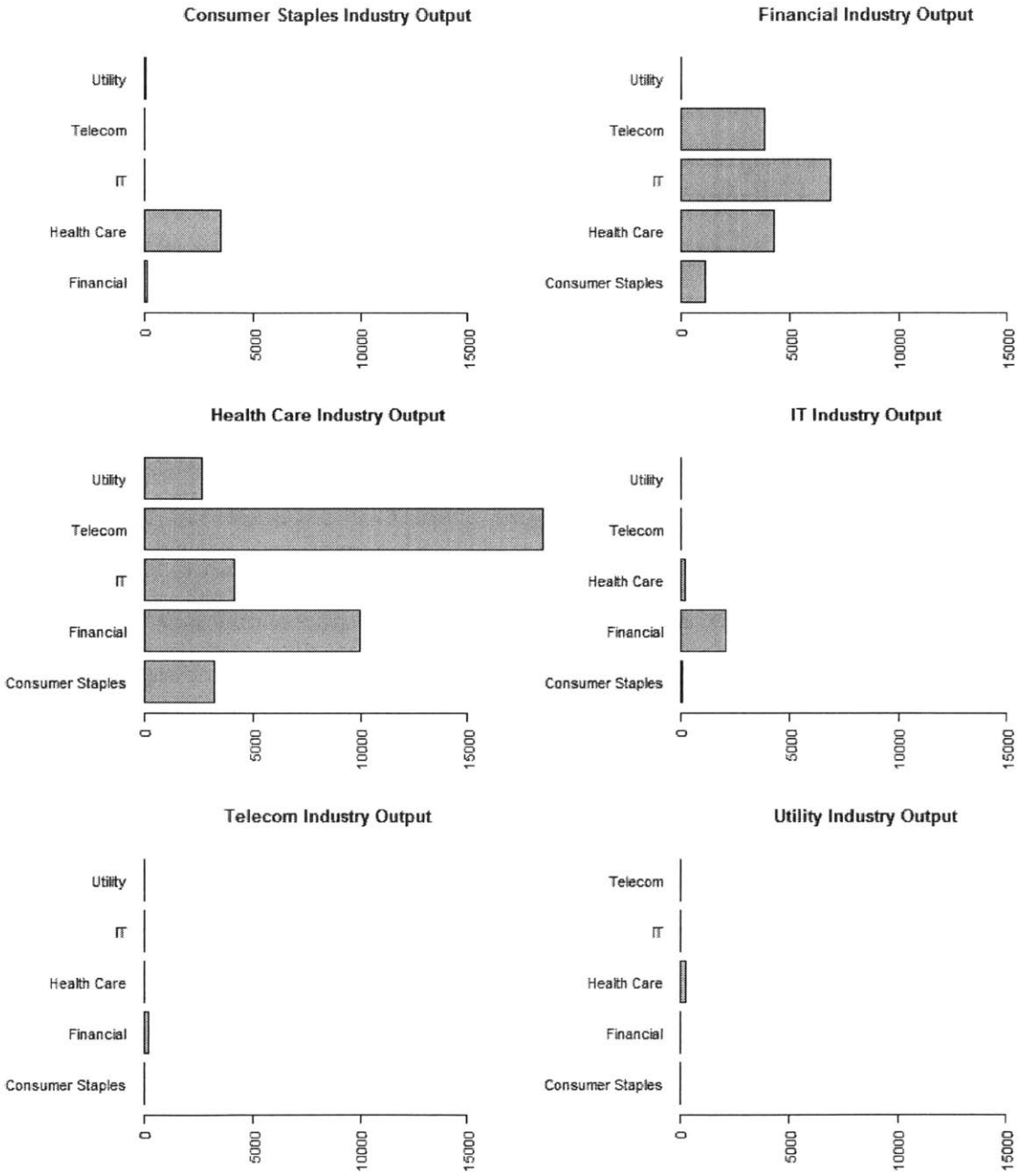
we have gained a greater flexibility in the range of pair-wise distributions that we can choose from.

Finally, we use the industry input-output matrix, which is published by the Bureau of Labor Statistics, as a quick validation for the vine dependence structure that we found in Figure 3.7.2.

Figure 3.7.3 shows the outputs (in millions of dollars) from the six industries we selected from 2008 to 2012, and the outputs are measured in 2005 chain weighted real dollars, where the chain weighted real dollar considers product substitutions made by consumers and other changes in their spending habits when measuring the CPI. Compared to the outputs, it is clear that the vine dependence structure that we obtained in Figure 3.7.2 has successfully uncovered several dominant input-output dependence relations: 1) Health Care and Telecom (with the highest output); 2) Financial and IT (with 3rd highest output); 3) Financial and Telecom. Therefore, we are confident that our vine dependence structure provides a reasonable underlying network/dependence structure among these six industries.

Note that the input-output matrix uses the industry output data, while our vine dependence structure was created using stock return data, and therefore the input-output matrix only serves as a tool to check the reasonability of our first level tree. We are by no means to reproduce the input-output matrix from the vine dependence structure, and vice versa.

Figure 3.7.3 - Output for the Six Industries (in millions of dollars)



3.8. Discussion

This chapter introduced a new robust technique for the vine selection process. This method minimizes the entropy function of the central piece of the data, and it preserves the breakdown points from the pairwise robust partial correlations. We also proposed an approximation heuristic using the maximum spanning tree algorithm that speeds up the runtime for finding a vine structure without much optimality gap, and such a heuristic has the property of permutation equivariance. Then, the Barrow Wheel benchmark shows that this search heuristic effectively captures the good portion of the data. For application, we extended the asset allocation application from the previous chapter by using the selected vine, and it delivers even better results than using the D-Vine structure. Also, we provide another application that utilizes the vine selection tool to uncover the underlying dependence/network structure between several industries, and it effectively transforms a high-dimensional modeling problem into simpler pairwise modeling problems.

Appendix 3.A. Demonstration of Algorithm 3.3.1

In this appendix, we will demonstrate by applying Algorithm 3.3.1 on the vine shown in Figure 3.3.1. The matrix representation of the vine is:

$$M^* = \begin{bmatrix} 4 & & & & & \\ 5 & 5 & & & & \\ 6 & 2 & 2 & & & \\ 2 & 1 & 6 & 6 & & \\ 1 & 3 & 3 & 1 & 1 & \\ 3 & 6 & 1 & 3 & 3 & 3 \end{bmatrix}$$

Rearranging the order of the variables to make the natural order to become (6,5,4,3,2,1), we have

$$\widetilde{M}^* = \begin{bmatrix} 6 & & & & & \\ 5 & 5 & & & & \\ 3 & 4 & 4 & & & \\ 4 & 2 & 3 & 3 & & \\ 2 & 1 & 1 & 2 & 2 & \\ 1 & 3 & 2 & 1 & 1 & 1 \end{bmatrix}$$

Step #	ρ_{a_1,a_n}	$(a_1, \ldots, a_n)$
1	$\rho_{2,1}$	(2,1)
2	$\rho_{3,1}$	(3,1)
3	$\rho_{3,2}$	(3,1,2)
4	$\rho_{4,2}$	(4,2)
5	$\rho_{4,1}$	(4,2,1)
6	$\rho_{4,3}$	(4,1,2,3)
7	$\rho_{5,3}$	(5,3)
8	$\rho_{5,1}$	(5,3,1)
9	$\rho_{5,2}$	(5,1,3,2)
10	$\rho_{5,4}$	(5,2,1,3,4)
11	$\rho_{6,1}$	(6,1)
12	$\rho_{6,2}$	(6,1,2)
13	$\rho_{6,4}$	(6,2,1,4)
14	$\rho_{6,3}$	(6,4,2,1,3)
15	$\rho_{6,5}$	(6,3,4,2,1,5)

At each step, $(a_1, \ldots, a_n)$ corresponds to Theorem 3.3.2, and one can easily check that the correlations in $r_1'(a_1, \ldots, a_n)$, $r_3'(a_1, \ldots, a_n)$ and $R_2(a_1, \ldots, a_n)$ are all known from previous steps, and therefore Theorem 3.3.2 can be applied to calculate ρ_{a_1,a_n}.

Appendix 3.B. Importance of Vine Selection

In this appendix, we will show, by an example, that it is very important to perform the vine selection process instead of assigning an arbitrary vine structure for the correlation estimation procedure.

Example 3.B.1. Let X_1, X_2, X_3 be three random variables that jointly follow a mixture of multivariable normal distributions.

Specifically, $(X_1, X_2, X_3) \sim \frac{1}{2}N_1 + \frac{1}{4}N_2 + \frac{1}{4}N_3$, where

- N_i is multivariate normal with mean $\vec{0}$, and covariance matrix Σ_i
- $\Sigma_1 = \begin{bmatrix} 1 & -0.9 & 0.81 \\ -0.9 & 1 & -0.9 \\ 0.81 & -0.9 & 1 \end{bmatrix}$
- $\Sigma_2 = \begin{bmatrix} 1 & 0.9 & -0.81 \\ 0.9 & 1 & -0.9 \\ -0.81 & -0.9 & 1 \end{bmatrix}$
- $\Sigma_3 = \begin{bmatrix} 1 & -0.9 & -0.81 \\ -0.9 & 1 & 0.9 \\ -0.81 & 0.9 & 1 \end{bmatrix}$

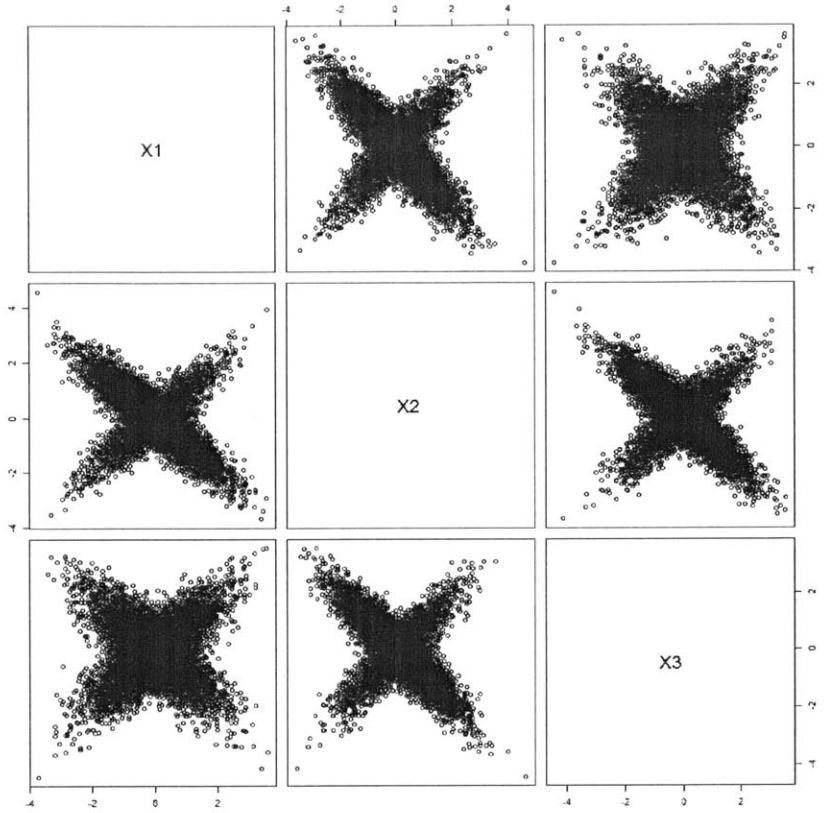
In Example 3.B.1, the central data is N_1, and the remaining ,i.e. N_2 and N_3, is noise/outlier.

Since we are doing pairwise estimation at a time, let's inspect the pairwise joint distributions first. This pairwise investigation will reveal a more natural vine structure for the estimation procedure.

It is easy to derive and plot the pairwise joint distributions for X_1, X_2, X_3:

- $(X_1, X_2) \sim \frac{3}{4}N\left(\vec{0}, \begin{bmatrix} 1 & -0.9 \\ -0.9 & 1 \end{bmatrix}\right) + \frac{1}{4}N\left(\vec{0}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}\right)$
- $(X_2, X_3) \sim \frac{3}{4}N\left(\vec{0}, \begin{bmatrix} 1 & -0.9 \\ -0.9 & 1 \end{bmatrix}\right) + \frac{1}{4}N\left(\vec{0}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}\right)$
- $(X_1, X_3) \sim \frac{1}{2}N\left(\vec{0}, \begin{bmatrix} 1 & -0.81 \\ -0.81 & 1 \end{bmatrix}\right) + \frac{1}{2}N\left(\vec{0}, \begin{bmatrix} 1 & 0.81 \\ 0.81 & 1 \end{bmatrix}\right)$

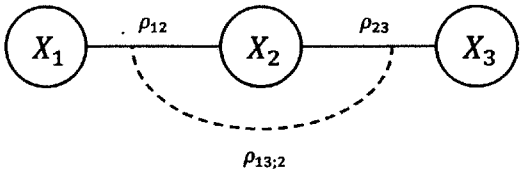
Figure 3.B.1 - Pairwise Scatter Plots



By inspection, (X_1, X_2) and (X_2, X_3) pairs clearly have less proportion of contamination, and hence robust estimators have a higher chance of getting the correct central piece. On the other hand, it is difficult to determine for the (X_1, X_3) pair which part is central data and which part is contamination because the split is 50/50.

Using simulation, it is confirmed that our intuition is correct, and we have to optimal vine structure shown in Figure 3.B.2.

Figure 3.B.2 - Optimal Vine Structure



If we incorrectly use any other vine structure, the first level tree will have the edge (X_1, X_3), and therefore we have to estimate ρ_{13} robustly. However, the robust estimates for ρ_{13} is very unstable due to the underlying joint distribution of (X_1, X_3). The robust estimator would give a positive correlation estimate approximately 50% of the time and a negative correlation estimate approximately 50% of the time as well, and such instability in the estimation procedure is highly undesirable, which leads to unstable final correlation matrix estimates. Figure 3.B.3 provides an example of such an incorrect vine structure, and we perform a comparison of the estimated ρ_{13} under these two vine structures in Table 3.B.1.

Figure 3.B.3 - Incorrect Vine Structure

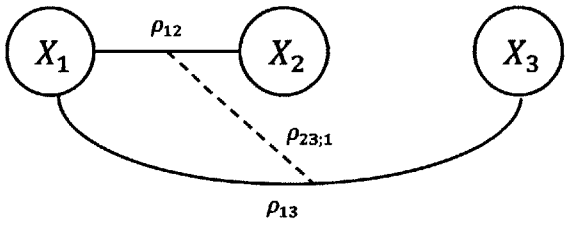
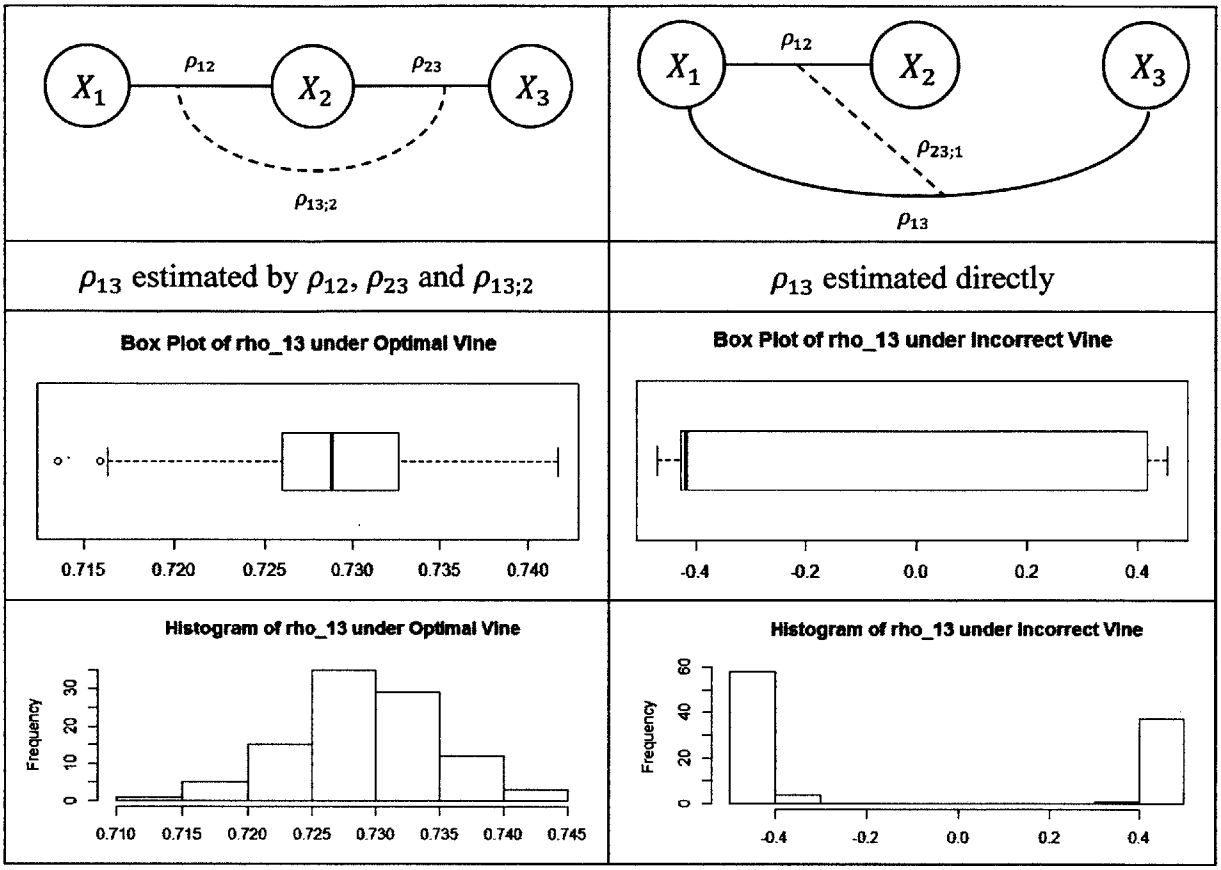
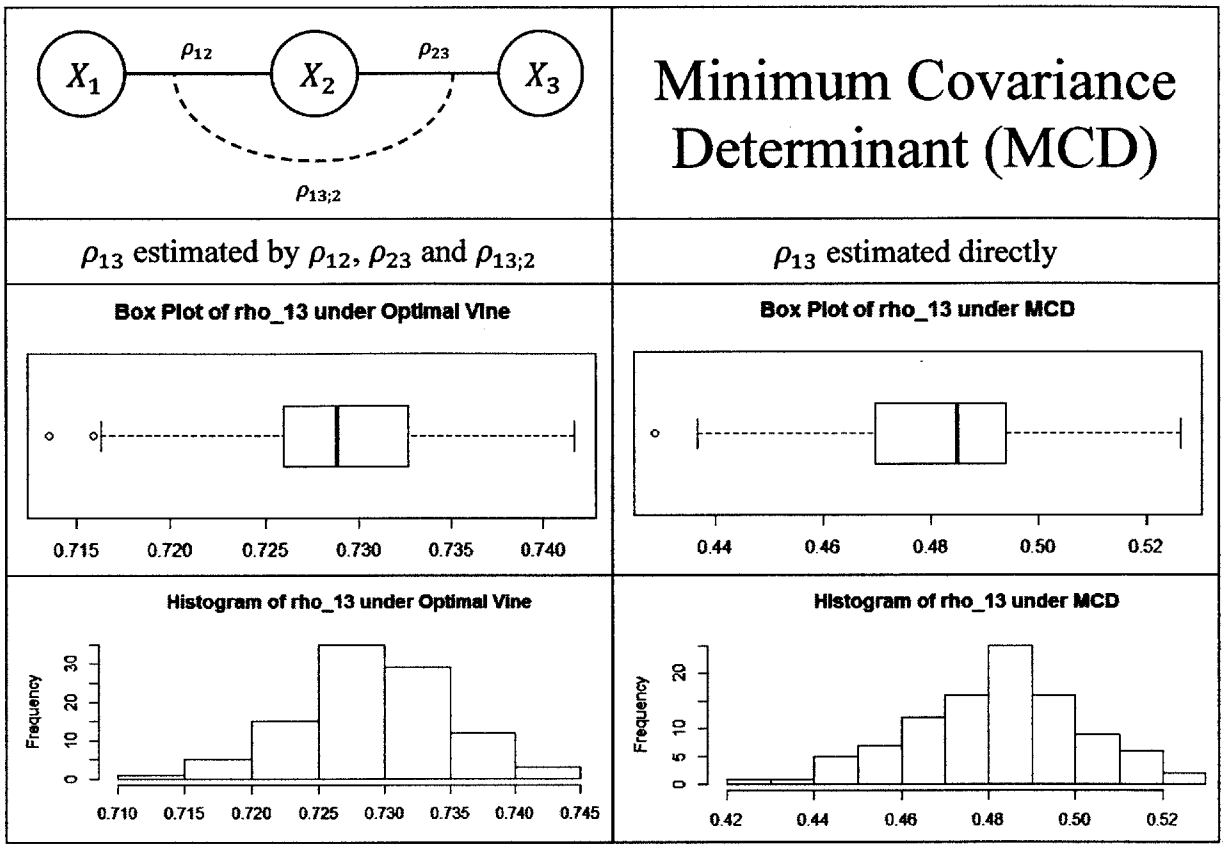


Table 3.B.1 - Comparison Optimal Vine Structure with Incorrect One



Finally, we show that global robust methods, such as Minimum Covariance Determinant (MCD), do not get confused by ambiguity by such pairwise distribution construction, and this is because they consider all variables while estimating instead of myopically looking at two variables at a time. Table 3.B.2 below provides a comparison of the estimated ρ_{13} under the optimal vine structure and MCD.

Table 3.B.2 - Comparison Optimal Vine Structure with MCD



Therefore, it is very important to identify the correct/optimal vine structure before proceeding with pairwise correlation estimations. The vine selection procedure identifies pairs with most prominent correlation/dependence, and it excludes any edge with low, and possibly ambiguous, correlation/dependence. In conclusion, the vine selection procedure helps on better estimating the correlation matrix.

Chapter 4

Statistical Learning for Variable Annuity Policyholder Withdrawal Behavior

In this chapter, we will analysis policyholder withdrawal behavior for the variable annuity product using modern statistical learning techniques.

4.1. Introduction

A variable annuity (VA) contract provides benefits that vary according to the investment experience of the assets supporting the contract. These assets may be segregated from the general assets of the insurance company in a separate account. Unlike traditional fixed annuity products issued by the insurance companies, the investment performance for VA depends on the market and not on the company's credited rate. The policyholders have the choice to allocate the VA investment into different funds, such as stock, bond, balanced and money market, and they get those funds' returns less any charges imposed by the insurance companies. In addition, policyholders have the option to purchase various guarantees for extra costs on top of the vanilla VA contract, and most common guarantees include Guaranteed Minimum Death Benefit (GMDB) and Guaranteed Minimum Withdrawal Benefit (GMWB).

The sales of VA products have been phenomenal over the past decades before the financial crisis, and this is because of its attractive features suitable for retirement financial planning purposes. By Towers Watson VALUE^{TM} Survey (Briere-Giroux, Huet, Spaul, Staudt and Weinsier, 2010), VA assets had exceeded $1 trillion by the end of 2003. During the

financial crisis, most VA sellers experienced huge losses on their VA lines, and sales have been limited by insurance companies for a while. However, in 2011, VA sales have again been strong and reached pre-crisis levels according to a report released by the Insured Retirement Institute (IRI) in March 2012, and the variable annuity net assets reached an all-time high of $1.61 trillion during the first quarter of 2012. Therefore, because of the significant position in the existing VA contracts together with their great potential for future sales and their resemblance to potential retirement solutions, it is of importance to study this product, and in particular, its policyholder behavior as this is one of the most important drivers of profit.

Insurance companies profit from VA products by charging various fees to policyholders, so policyholder behavior has a great impact on the company's profit streams. Common charges include surrender charges, asset management charges, guarantee charges and annual fees. Surrender charges occur when policyholders prematurely surrender their VA contracts and cash out their funds from the insurance company, and it is usually a certain percentage of the total account value, and this percentage varies by the duration of the VA contracts. For example, the insurance company may impose a 7% surrender charge in the first year of the contract, and then 6% in the second year, and 5% in the third year, and so on until there is no surrender charge in the eighth year and beyond. Asset management charges in the form of an annual fee, ranging from 1% to 3% of the fund value, are collected by the insurance company for providing the asset management service. Guarantee charges are applied when any of the guarantees are purchased with the VA contract, and they are a specified percentage of the fund value depending on the guarantee type. Finally, insurance companies usually charge $25-$35 annual fees for administrative purposes.

Insurance companies typically experience losses when policyholders surrender or annuitize their contracts. In the event of a full surrender or annuitization, insurance companies lose all future streams of income described in the paragraph above. Also, if some form of surrender value guarantee is effective at the time of surrender and is higher than the fund value, insurance companies incur a loss equal to the difference between the guarantee

amount and the fund value. Therefore, it is apparent that policyholder behavior is one of the most important profit and loss drivers for the VA business.

There are generally two types of variable deferred annuity contracts: single premium deferred annuity (SPDA) and flexible premium deferred annuity (FPDA). SPDA, as specified by its name, allows only one premium payment at the beginning of the VA contract and no future premiums. FPDA, on the other hand, allows a periodic premium to be paid for the VA contract, though the majority policyholders would pay a very large lump sum at the beginning and relatively small payments after that. In this chapter, we will only consider the SPDA case, and the effect of future premiums on lapse is not analyzed.

4.2. Current Practice and Research

There are three types of policyholder behaviors that have received the most focus under the current practice of insurance companies:

1. Dynamic lapse
2. Partial withdrawal
3. Annuitization.

Dynamic lapse addresses the issue when policyholders surrender their VA contracts for cash before the specified maturity date. It is well understood in the insurance industry that the lapse rate is a function of both the base behavior and the dynamic behavior. The base withdrawal behavior is affected by the characteristics of the policyholder and the contract features, while the dynamic withdrawal behavior is caused by the performance of the contract, e.g. in-the-moneyness of the living benefit guarantees. The insurance industry knows about the qualitative relationship between various factors and the withdrawal rate. For example, low withdrawal penalty and/or low guarantee amount for the benefit are associated with a high withdrawal rate. However, there is not a sophisticated quantitative model for such associations at the current time.

According to the Towers Watson paper published in 2010, which is one of the latest published research papers we could find on this topic, traditional quantitative models typically use a simple approach incorporating the following factors:

- Base lapse rate
- Surrender charge and its structure (e.g. dollar for dollar provision)
- Lapse at the end of the surrender charge period
- Commission structure
- In-the-moneyness of the guarantees.

One of the major shortcomings of such an approach, as pointed out in the Towers Watson paper, is that the traditional models do not fully account for correlations between the explanatory variables and they do not use other readily available variables, such as age and gender. More details on the traditional approach are covered in the Towers Watson paper.

The Towers Watson paper considers using the generalized linear model to capture the dependency among explanatory variables. The Towers Watson paper obtains more favorable results compared to traditional approaches.

Another research paper published by Milliman (Sun, 2011) also discussed the data-mining approach for the variable annuity dynamic lapse study. This chapter also only considered linear model in their demonstrated example. However, given advances in the field of statistical learning and modeling, there is still room for possible improvement as this chapter will demonstrate.

Partial withdrawal addresses the issue when a policyholder withdraws a portion of the fund value without fully surrendering the VA contract. There are several considerations for partial withdrawal. One of the most important is the dollar-for-dollar withdrawal provision in the presence of any Guaranteed Minimum Benefit. Without such a provision, if a policyholder performs a partial withdrawal, the Guaranteed Minimum Benefits will be

reduced by the ratio of the amount withdrawn over the original fund value. However, with a dollar-for-dollar withdrawal provision, the Guaranteed Minimum Benefits will be reduced in dollar amount equal to the amount withdrawn, and policyholders can be strategic in the presence of such a provision. For example, if the fund value is $75 and the guarantee is $100, which is a possible scenario after the financial crisis, the fund has to grow 33% to reach the guarantee. If the policyholder withdraws $25 at this time, the fund value will be reduced to $50, and the guarantee will be reduced to $75, and the fund has to grow 50% to reach the new guarantee. This phenomenon is even more amplified as the guarantee is deeper in the money, which corresponds to the situation at the current post-crisis time. Another important consideration for partial withdrawal is the presence of GMWB, which allows the annuitant to withdraw a maximum percentage of fund value each year until the initial investment amount has been recouped. Policyholders can act strategically and exercise this option during a time of poor investment returns, and this applies usually to wealthy policyholders with large fund values because they have access to sophisticated financial advisors.

The last item is the study of annuitization, and the considerations are similar to that for dynamic lapse. There is one important factor in annuitization that is not considered in dynamic lapse, and it is the presence of GMIB, which guarantees the minimum payments if the policyholder chooses to annuitize. As in-the-moneyness of GMIB increases, annuitization rates increase. However, such a relationship is generally not linear as shown in Scotchie's paper (2006).

In this chapter, we will focus on modelling for dynamic lapse, and the other two topics of interest, namely partial withdrawal and annuitization, can be modelled with similar approaches should we have the relevant data. We will first apply the logistic regression model, which is used in the Towers Watson paper, to the data, but with a modern feature selection technique. We will then introduce recent machine learning techniques for classification. The majority of the analyses are concentrated in these sections because they

are most relevant to the valuation process and have not been studied in the academic literature.

4.3. Data

The data used in the analysis resembles common characteristics of an actual insurance company that sells VA contracts. The data included the lapse experience during the recent financial crisis.

The following variables, with variable names in the brackets, are included in the data before the variable selection process:

- Write company: two subsidiary companies that write VA business
- Product group: variable with 5 categories
- Plan type: Qualified and Non-qualified
- Version: 1 to 8
- Special program indicator: Y or N
- Withdrawal option: variable with 6 categories
- Commission type: variable with 7 categories
- Dollar-for-dollar withdrawal indicator: Y or N
- Account value: numerical
- Death benefit: numerical
- Total benefit: numerical
- Withdrawal benefit: numerical
- Attained age: variable with 20 categories
- Feature duration: 1 to 20
- Indicator if in cliff year: variable with 3 categories
    - The cliff year is the final year with withdrawal penalties
- Grouping for net amount at risk: variable with 8 categories

- Grouping for account value: variable with 7 categories
- Grouping for in-the-moneyness/out-of-the-moneyness: variable with 9 categories for different ranges of moneyness.
- Indicator for in-the-moneyness/out-of-the-moneyness: variable with 3 categories
- Indicator whether the contract is surrendered: Y or N

Obviously, the last variable is the one that we are predicting, and all other variables are available for building models.

4.4. Traditional Logistic Model with Feature Selection

First, we use the traditional logistic model, but with feature selection to reduce model complexity. There may be unnecessary fields, either due to irrelevance to the prediction variable or high correlation with other explanatory variables, and by first doing feature selection, we are able to reduce the number of variables for prediction and thus reduce model complexity and possibly increase model prediction accuracy.

For the feature selection process, we use logistic regression with group lasso (Meier, 2007). Let $\boldsymbol{x_{i,j}} \in \mathbb{R}^{p_j}$ be the jth explanatory variable column vector for the ith observation, and p_j is the effective number of variables or the degrees of freedom for $\boldsymbol{x_{i,j}}$. For example, a categorical variable with 8 levels has 7 dummy indicator variables, so $p_j = 7$, and a continuous variable has $p_j = 1$. Let $\boldsymbol{\beta_j}$ be the corresponding coefficient vector. Denote $\boldsymbol{x_i} = \left(1, \boldsymbol{x'_{i,1}}, \ldots, \boldsymbol{x'_{i,p}}\right)'$, $\boldsymbol{\beta} = \left(\beta_0, \boldsymbol{\beta'_1}, \ldots, \boldsymbol{\beta'_p}\right)'$, and then the log-likelihood function for standard logistic regression with two classes is:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i \boldsymbol{\beta}' \boldsymbol{x_i} - \ln\left(1 + e^{\boldsymbol{\beta}' \boldsymbol{x_i}}\right)\right)$$

The logistic regression with group lasso is to minimize the following function over β:

$$f(\boldsymbol{\beta}) = -l(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \sqrt{p_j} \|\boldsymbol{\beta}_j\|_2$$

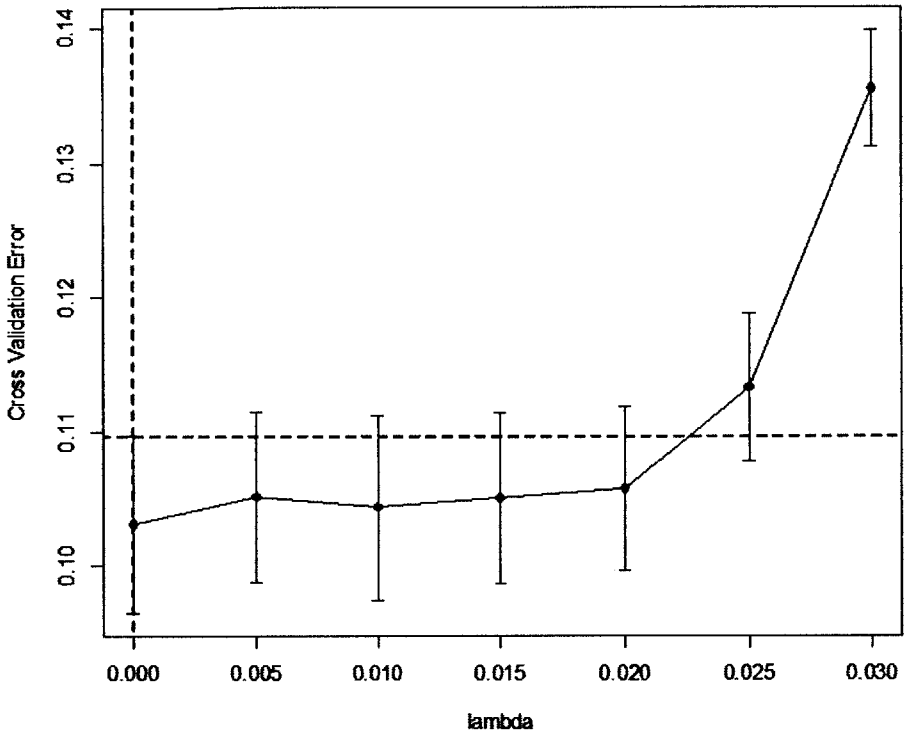
If all variables are continuous, then $p_j = 1$ and β_j's are of single dimension for all j, and the penalty term can be expressed as:

$$\lambda \sum_{j=1}^{p} \sqrt{p_j} \|\beta_j\|_2 = \lambda \sum_{j=1}^{p} \sqrt{1} \cdot \sqrt{\beta_j^2} = \lambda \sum_{j=1}^{p} |\beta_j|$$

This corresponds to the l_1 penalty term used in the lasso. However, since there are typically several dummy indicator variables corresponding to a single categorical variable, we would prefer to have a feature selection process where, for every categorical variable, the coefficients for the corresponding dummy variables are either all zero or all non-zero. The group lasso provides exactly this feature of leaving a categorical variable in or out entirely, instead of leaving some of its dummy variables in and some out. The nuisance parameter $\lambda \geq 0$ controls the size of the penalties and, in effect, the number of parameters in the final model. We choose λ, over the range from 0 to 0.03, by 10-fold cross validation, and the lowest average misclassification error is at $\lambda = 0$, but all $\lambda \leq 0.02$ have average misclassification falling within one standard deviation of the $\lambda = 0$ case, so by the law of parsimony, we choose $\lambda = 0.02$. Variables with zero coefficients are dropped.

Therefore, the final selected explanatory variables are version, withdrawal option, commission type, dollar-for-dollar withdrawal indicator, feature duration, and indicator if in a cliff year. The logistic regression model with lasso feature selection has an average misclassification rate of 10.58% with a standard error of 0.58%.

Figure 4.4.1 - Logistic Regression with Group Lasso: 10-fold Cross Validation



4.5. Machine Learning for Individual Policyholder Withdrawal Modelling

In this section, we consider various modern machine learning techniques, including Naïve Bayes, Bayesian Network, K-Nearest Neighbor, Decision Tree using CART, C4.5 Tree, Bagging, Boosting, Random Forest and Rotation Forest. Since most of the explanatory variables are categorical, we should expect methods like Bayes and decision trees to perform better than methods like nearest neighbors.

The focus is on the Rotation Forest (Rodriguez and Kuncheva, 2006), a relatively new machine learning method that performs rather well for this dataset.

Naïve Bayes

The Naïve Bayes classifier is the simplest machine learning algorithm, and it is the first one we apply to the data. The classifier assumes that given a certain class, the features are independent. Over 10-fold cross validation, the misclassification rate has a mean of 24.30% and a standard error of 0.63%. We will drive down the misclassification rate using other classifiers.

Bayesian Network Classifier

The Naïve Bayes classifier assumes a rather simple dependence structure among variables, and this dataset has the property of having many more data points compared to explanatory variables, so it may be possible to exploit this characteristic of the dataset to build a more sophisticated, and hopefully, better dependency structure for the modelling. One such approach is the Bayesian Network Classifier (Friedman, Geiger & Goldszmidt, 1997). See Appendix A for details on the Bayesian Network Classifier with graphical demonstrations.

Applying the Bayesian Network classifier using the Tree-Augmented Naïve Bayes (TAN) algorithm (Friedman, Geiger & Goldszmidt, 1997) under 10-fold cross validation, we achieve an average of 10.80% misclassification rate with a standard deviation of 0.66%, which is a significant improvement over the Naïve Bayes method, and it is almost as good as the traditional logistic regression considered previously ($10.58\% \pm 0.58\%$). Also, the algorithm for running the Bayesian Network is really fast (up to 5x faster than logistic regression), which makes the Bayesian Network a perfect substitute for Naïve Bayes as a preliminary tool for classification analysis.

Decision Tree using CART

The first decision tree classifier we consider is a learner using CART's minimal cost complexity pruning, and we set it to use 5-fold cross validation for the pruning process. This decision tree algorithm only allows binary splits, and we will relax this when we consider the next decision tree classifier. Over the 10-fold cross validation, it achieves an average of 9.51% misclassification rate with a standard deviation of 0.65%, lower

compared to the Bayesian Network classifier and better than logistic regression within one standard deviation.

C4.5 Tree

C4.5 Tree is another decision tree classifier, and different from using CART, it allows multiple splits at each parent node, thus allowing greater flexibility for the model. However, such flexibility does not come without a cost. Over the 10-fold cross validation, it performs worse than that using CART 7 out of 10 times, and it has an average of 9.64% misclassification rate with a standard deviation of 0.66%. Therefore, on average, C4.5 performs worse than CART, but the runtime is faster using C4.5 Tree than CART.

K-Nearest Neighbor (kNN)

The kNN method does not perform well for this dataset compared to the Bayesian Network, Decision Tree using CART, or the C4.5 Tree. The reason is because there is a mixture of categorical and numerical variables in the dataset, and the traditional choice of distance functions, such as Euclidean distance, does not generally perform well.

We ran the kNN classifier with different numbers of k, ranging from 1 to 25, and the best performance is achieved with k=17 with an average misclassification rate of 11.72% with a standard deviation of 0.77% over 10-fold cross validation. This classifier performs worse than the logistic regression model and is not considered for further analysis.

Bagging

We now move the focus to ensemble methods, starting with Bagging. We use the decision tree with the reduced error pruning algorithm as the base classifier for the Bagging ensemble classifier. The reduced error pruning algorithm replaces the nodes, starting from the leaves, with its most popular class, and it checks if the prediction accuracy is affected. It is faster than the normal pruned decision tree algorithms and better than the decision stump, so we decide to use it as the base classifier.

We ran the Bagging algorithm with different numbers of iterations, ranging from 50 to 150, and the algorithm with 50 iterations gave the best performance with an average of 10.22% misclassification rate with a standard error of 0.73% over the 10-fold cross validation.

Boosting

We use the decision stump as the base classifier for the Boosting ensemble classifier because it is a simple base classifier and works rather well. We run the Boosting algorithm with different numbers of iterations, ranging from 50 to 150, and the algorithm with 125 iterations gives the best performance of an average of 9.86% misclassification rate with a standard error of 0.66% over the 10-fold cross validation. Therefore, Boosting gives a rather good result.

Random Forest

The Random Forest classifier bootstraps a specified number of samples and builds a random tree on each sample. While building a random tree, the classifier randomly selects a subset of the features to consider at each split point, and we choose the size of such a subset to be $\ln(p) + 1$, where p is the number of features. Such randomization is able to make the classifier diverse without sacrificing too much performance. We ran the Random Forest classifier over different numbers of random trees, ranging from 50 to 200. Over the 10-fold cross validations, the best average misclassification rate of 11.12% with a standard deviation of 0.73% is achieved with 125 trees, and it does not perform well.

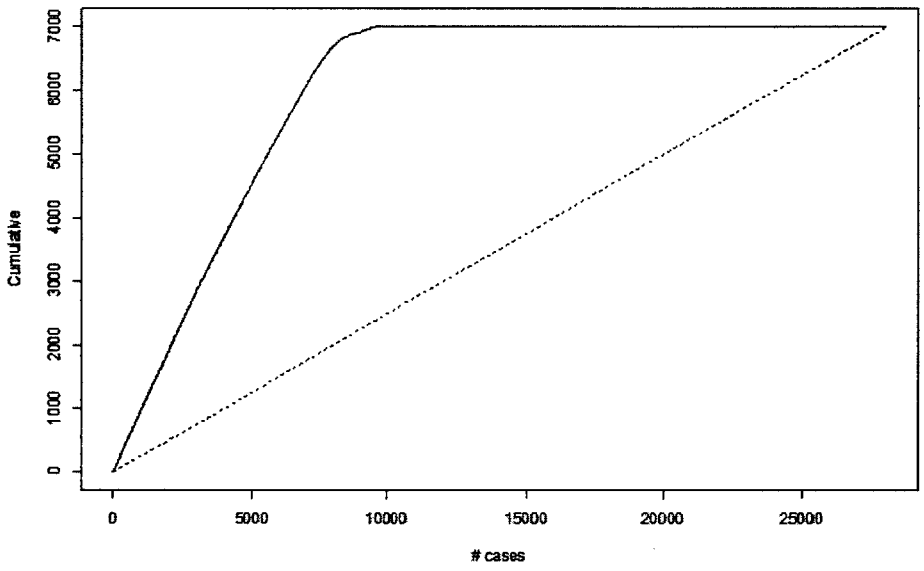
Rotation Forest

Rotation forest is a relatively new ensemble classifier, and it was initially published in 2006 by Rodriguez and Kuncheva. See Appendix B for details on the Rotation Forest Classifier.

Empirical experiments have shown that the rotation forest classifier can give similar or better performance compared with the random forest classifier. This has been proven for our dataset as well, and, as we will demonstrate in the Overall Performance Comparison section, Rotation Forest performs better than any other method for 9 out of 10 folds of cross validation.

Over 10-fold cross validation, we run the Rotation Forest classifier with different numbers of iterations, ranging from 50 to 200, and we compare the average misclassification rates. We find out that the average misclassification rate achieves the minimum of 9.26% with 125 iterations over 10-fold cross validation with an associated standard error of 0.70%. To produce the lift chart, we use the entire training data and build the Rotation Forest classifier with 125 iterations, and then we plot the lift chart below. Visually, in Figure 4.5.1, we can see that there are significant gains from using the Rotation Forest classifier, and the classifier indeed gives good performance.

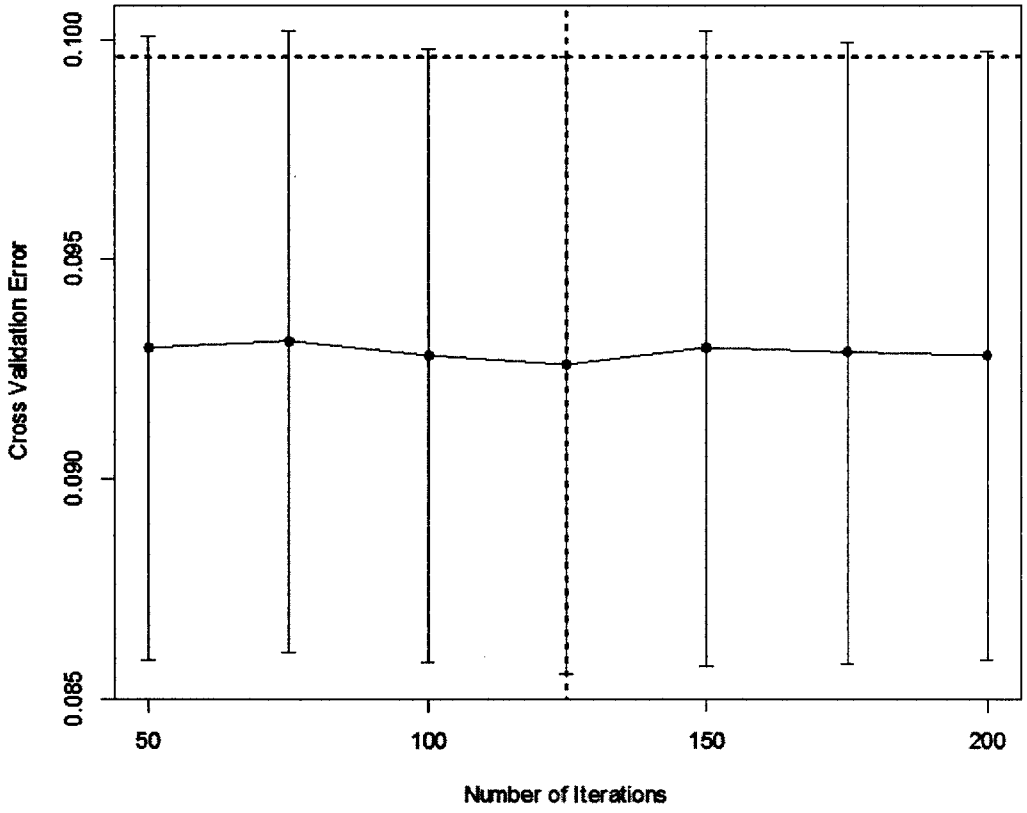
Figure 4.5.1 - Lift Chart: Rotation Forest (with 125 iterations)



Although the average misclassification rate achieves the minimum with 125 iterations, it may be caused by statistical fluctuations. There may be a Rotation Forest with fewer iteration runs that delivers similar performance but has a higher average misclassification rate for this dataset simply due to random fluctuations. Therefore, we compute the range of misclassification rates within one standard deviation for the classifier with 125 iterations, and we discover, as shown in Figure 4.5.2 below, that even with different iterations, all

Rotation Forest classifiers' average misclassification rates fall within the range, so there is actually no statistical difference in performance using different iteration counts. Therefore, in practice, we can choose the model with the lowest complexity, i.e. the Rotation Forest with 50 iterations.

Figure 4.5.2 - Rotation Forest: 10-fold Cross Validation



Overall Performance Comparison

We measure the performance of a classifier using the average misclassification rate over the 10-fold cross validation, and we take into consideration the standard deviation of the misclassification rates over the cross validation.

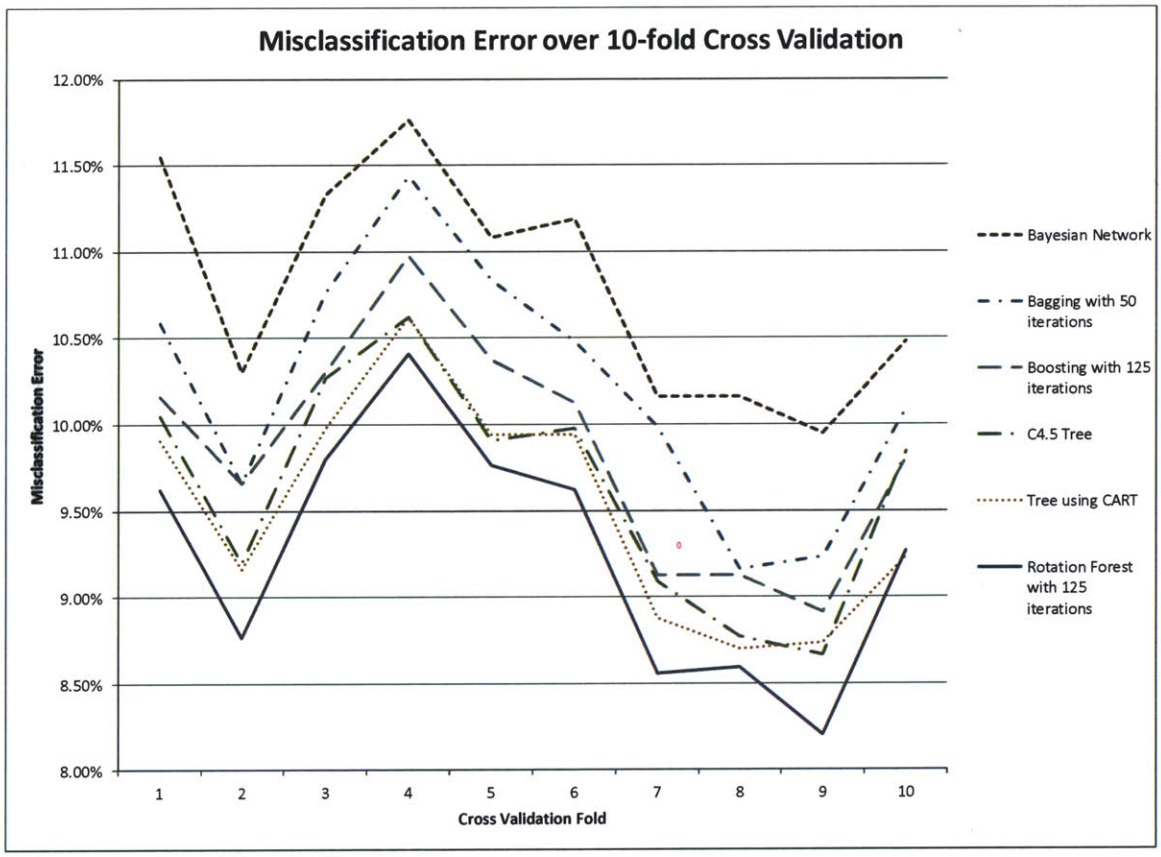
Table 4.5.1 provides a summary of comparisons between the classifiers together with comments on their performance. The model parameters, such as the k in kNN and the number of iterations in ensemble methods, were chosen over 10-fold cross validation. The best model, in terms of misclassification error, is the Rotation Forest classifier, and the best non-ensemble method is the Decision Tree using CART.

Table 4.5.1 - Summary of Model Performances

Method	Misclassification Rate		Ensemble Method?	Comments
	Mean	Standard Error		
Logistic Regression	10.58%	0.58%	No	Baseline model
Naïve Bayes	24.30%	0.63%	No	Misclassification rate is much higher than the baseline model
Bayesian Network	10.80%	0.66%	No	Up to 5x faster compared to logistic regression; good for preliminary analysis
Decision Tree using CART	9.51%	0.65%	No	Best non-ensemble method in terms of misclassification rate
C4.5 Tree	9.64%	0.66%	No	Second best non-ensemble method with faster runtime compared to CART
kNN (k = 17)	11.72%	0.77%	No	Misclassification rate is above one standard error of the baseline model, i.e. logistic regression
Bagging (50 Iterations)	10.22%	0.73%	Yes	Misclassification rate is above one standard error of the best model, i.e. Rotation Forest
Boosting (125 Iterations)	9.86%	0.66%	Yes	Second best ensemble method with faster runtime compared to Rotation Forest
Random Forest (125 Iterations)	11.12%	0.73%	Yes	Misclassification rate is above one standard error of the best model, i.e. Rotation Forest
Rotation Forest (125 Iterations)	9.26%	0.70%	Yes	Best model in terms of misclassification rate

Figure 4.5.3 shows the misclassification error over the 10-fold cross validation for each classifier. Again, over all the non-ensemble classifiers considered, Decision Trees using CART delivers the best performance, and the C4.5 Tree is very close. Over all the ensemble classifiers considered, Rotation Forest is the best performer. However, as shown in Figure 4.5.4 (left), CART, C4.5 and Boosting all fall within the one-standard-deviation bound. Therefore, we conclude that Rotation Forest has one of the best classification performances, but CART, C4.5 and Boosting have similar performance statistically.

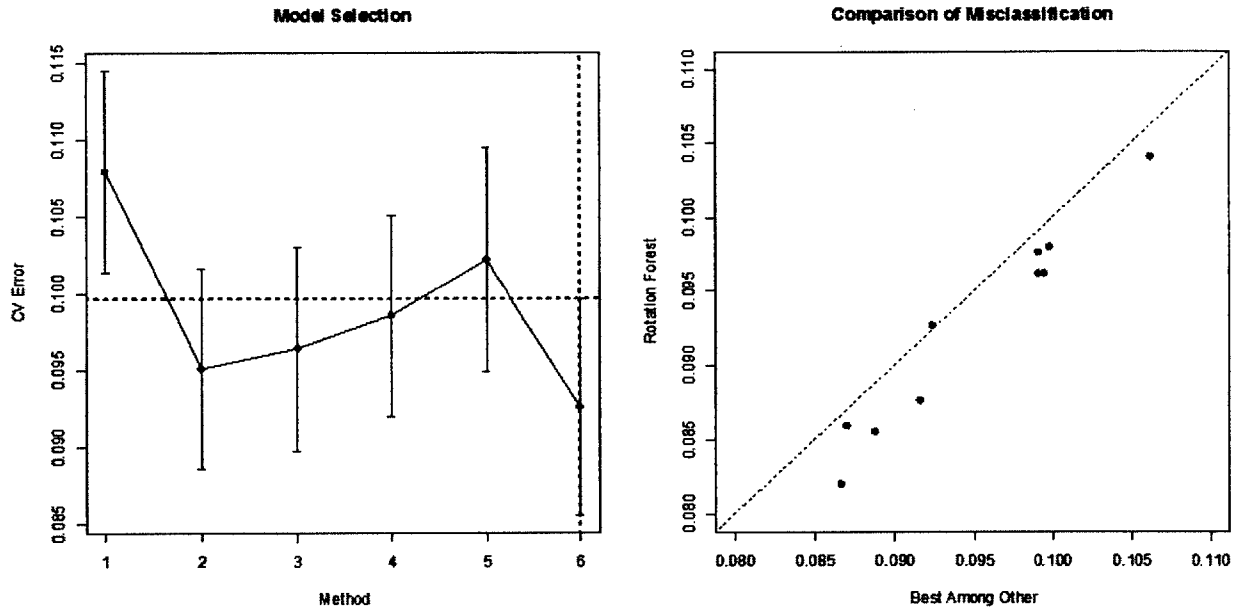
Figure 4.5.3 - Misclassification Error for all Models



We also include the figures showing the misclassification error for the classifiers over each of the 10 folds as well as the comparison, for each fold, between Rotation Forest and the best amongst all other classifiers as shown in Figure 4.5.4 (right). It is clear that Rotation

Forest stands out with 9 out of 10 folds outperforming the other classifiers and, in most cases, by a good margin.

Figure 4.5.4 - Comparison of Different Models



The methods in the left figure are: 1. Bayesian Network, 2. Decision Tree using CART, 3. C4.5 Tree, 4. Boosting (125 iterations), 5. Bagging (50 iterations), and 6. Rotation Forest (125 iterations)

There are other performance measures that may be more relevant within the business context, such as the cost of misclassification. Those analyses are fairly similar to ours and can be easily carried out.

Therefore, we would recommend using C4.5 Tree for initial exploration and as the basis for performance measures, and if there is a constraint on computation power or time, the C4.5 Tree classifier would suffice as the final classifier. However, if no such constraint is present, for example monthly valuation on high performance servers, we would recommend using the Rotation Forest classifier.

4.6. Discussion

In this chapter, we have applied statistical learning techniques to model the VA policyholders' dynamic withdrawal behaviors. Most methods applied are modern statistical models, and they deliver better results than the traditional approach used by many insurance companies today. Also, the machine learning techniques applied to model individual VA policyholder behavior are able to give prediction on a policy-by-policy basis.

Our dataset is a rather good representation of the ones in the insurance industry: there are many categorical features, and the number of data entries is significantly more than the number of features. Therefore, our analysis can be applied to other insurance data analysis, for example, VA annuitization rate studies as well as general mortality studies.

The feature selection process, which has been carried out in section 4.4, uses logistic regression as the base model. Future research remains to compare feature selection done independently with each machine learning method described in section 4.5.

Also, the kNN classifier with Euclidean distance measure did not perform well on this dataset. Future research remains to compare classification performances using different distance measures.

Appendix 4.A. Details on Bayesian Network Classifier

A Bayesian Network is an acyclic graphical model incorporating a set of random variables and their conditional dependency structure using directed edges. To use it as a classifier, we make the assumption that given its parents, each node in the graph is conditionally independent of any other set of non-descendent nodes, or mathematically:

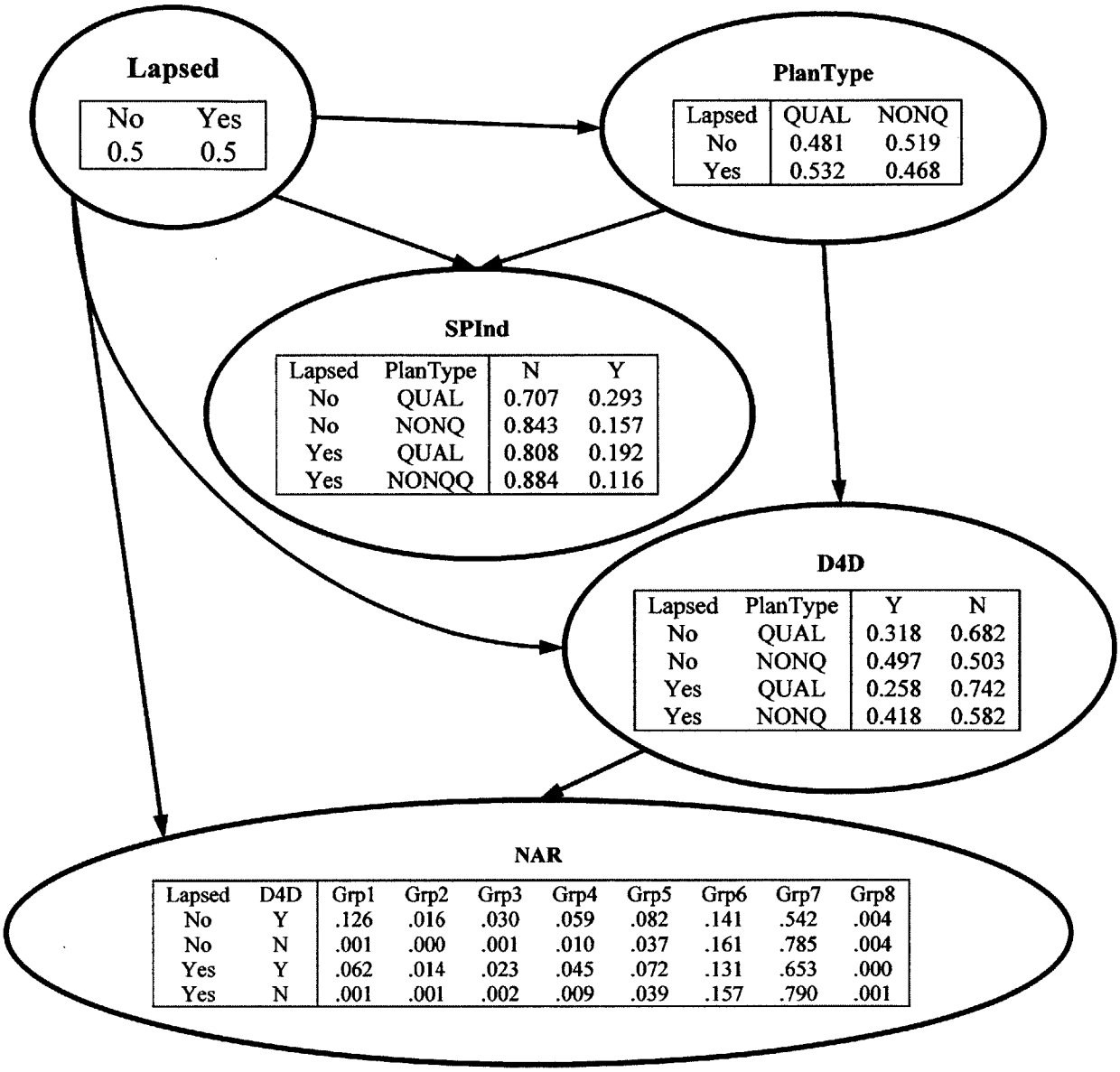
$$P(node|parents\ \&\ other\ non-descendants) = P(node|parents)$$

Because the Bayesian Network model is an acyclic graph with the prediction variable y as the parent node, the explanatory nodes can be ordered in such a way that all ancestors of a node x_i have indices smaller than i, and applying the multiplicative rule of conditional probability:

$$P(y, x_1, \dots, x_n) = P(y) \cdot \prod_{i=1}^{n} P(x_i|y, x_{i-1}, \dots, x_1) = P(y) \cdot \prod_{i=1}^{n} P(x_i|x_i's\ parents)$$

For example, we might use the following four explanatory variables into our model: plan type (PlanType), special program indicator (SPInd), dollar-for-dollar withdrawal indicator (D4D) and grouping for net amount at risk (NARgrp). A possible Bayesian network looks like the following figure with conditional probability tables included. The probabilities in a single row should always sum up to 1.

Figure 4.A.1 - Example of the Bayesian Network Model



The prediction variable, Lapsed, is the parent node of the network, and it has directed edges connecting to all the explanatory variables in the model, and then there are directed edges connecting the explanatory variables to represent the dependency structure among them. The table associated with each of the variables represents the conditional probability of each node given its parents. For example, the first number (0.126) in the table for the

NAR variable is the probability of NAR being "Grp1" given that Lapsed is "No" and D4D is "Y".

Given the tables, it is very convenient to calculate probabilities, and those calculated probabilities are used in classification by simply classifying the item to the category with the highest probability. For example, let A be the event that PlanType is "QUAL", SPInd is "Y", d4dInd is "Y", and NAR is "Grp5", then P(Lapsed = "No", A) = 0.481 × 0.293 × 0.318 × 0.082 = 0.0036750, and P(Lapsed = "Yes", A) = 0.532 × 0.192 × 0.258 × 0.072 = 0.0018974, so we would predict that this policyholder will not surrender the contract under the model.

We want to note the fact that the Naïve Bayes classifier is just a simple case of the Bayesian network with directed edges connecting only the prediction variable to each explanatory variable, and the property is exploited later for building more complex Bayesian networks from this simple structure.

It is obvious that a large set of Bayesian networks can be constructed using the same set of variables, and this presents two tasks when using such a classifier: a method to search through the space of possible networks and an evaluator to compare and select the best model. To search for the network, we use the Tree-Augmented Naïve Bayes (TAN) algorithm (Friedman, Geiger & Goldszmidt, 1997), which performs rather well on our data and empirically by other studies. To evaluate a given network, we use the likelihood value computed by the data and possibly apply penalties for more complex models, i.e. models with more parameters.

The TAN algorithm for building Bayesian networks takes the Naïve Bayes classifier and adds edges to it. In the Naïve Bayes network, there is only a single parent for each explanatory variable node, and TAN considers adding a second parent to each of those nodes. This algorithm gives the benefits of more complex dependence modelling than Naïve Bayes, and it generalizes well to new data. There is an efficient algorithm for

finding the set of edges to maximize the likelihood based on the maximum weighted spanning tree. The Bayesian networks built in the example above as well as in the final model are both found by the TAN algorithm.

For model selection, the common approach in statistics is using the log-likelihood together with a penalty term measuring model complexity. Let N be the number of data points, LL be the log-likelihood, and K be the number of parameters for the network. One popular measure is the traditional Akaike Information Criterion (AIC):

$$\text{AIC} = -LL + K$$

Another popular one is the Minimum Description Length (MDL), which is similar to the Bayesian information criterion (BIC):

$$\text{MDL} = -LL + \frac{\ln(N)}{2}K$$

After choosing an evaluator, the network with the minimum value of the measure is the winning one. To calculate the number of parameters in a network, we simply sum over the number of parameters in each conditional probability table. For example, in the network above, the total number of parameters is $1 + 2 + 4 + 4 + 28 = 39$.

Appendix 4.B. Details on Rotation Forest

The rotation forest is a relatively new ensemble classifier, and it was initially published in 2006 by Rodriguez and Kuncheva. At each iteration, it incorporates the characteristics of the random forest, bagging, and principal component analysis to generate decision trees, and it uses the average of the probabilities over all iterations and classifies the item to the category with the highest average probability.

The detailed process is described below for each iteration:

1. Divide the set of all p features (i.e. explanatory variables) randomly into K subsets. We choose disjoint subsets to maximize the chance for high diversity. Let $M_1,\ldots,M_K$ be the size of each subset, so $\sum_{j=1}^{K} M_j = p$.
2. For each subset $j \in \{1, \ldots K\}$ of the features, we draw a bootstrap sample using 50% of the original data, and we apply principal components analysis (PCA) using only the features in the subset to get linear combinations of the attributes in the subset. Denote the coefficients obtained in PCA as $a_j^{(1)}, \ldots, a_j^{(M_j)}$, each of size $M_j \times 1$.
3. Construct the "rotation" matrix R as:

$$R = \begin{bmatrix} a_1^{(1)}, \ldots, a_1^{(M_1)} & [0] & \cdots & [0] \\ [0] & a_2^{(1)}, \ldots, a_2^{(M_2)} & \cdots & [0] \\ \vdots & \vdots & \ddots & \vdots \\ [0] & [0] & \cdots & a_K^{(1)}, \ldots, a_K^{(M_K)} \end{bmatrix}_{p \times p}$$

4. Rearrange the columns of R so that they correspond to the original order of the features. Denote the rearranged rotation matrix as R^a.
5. Train the classifier in the ensemble, e.g. decision tree, using XR^a as the feature set.

Chapter 5

Contributions and Future Work

In this thesis, we made contributions in robust dependence/correlation modeling and applied various statistical techniques to insurance and retirement applications.

First, we introduced a new robust dependence/correlation modeling technique that incorporates both the regular vine structure and robust partial correlation modeling. This technique guarantees the positive definiteness of the estimated correlation matrix, preserves the breakdown point in the estimation process, and delivers good results in the benchmark while improving computing time for large datasets.

Then, we formulated the optimization problem for finding the optimal robust vine based on minimum entropy, and proposed a heuristic to solve this optimization problem by going through the vine level by level, and find the maximum spanning tree according to the robust correlation measure on each level.

On the application side, we applied the proposed technique to the asset allocation problem with investment assumptions suitable for insurance companies, and the allocation method using covariance matrix estimated by this new technique outperforms other popular asset allocation methods based on both accumulation and the information ratio. We also utilized the vine selection technique to find the underlying dependence structure between different industries, and this dependence structure allowed us to transform the multivariate problem into a series of bivariate analyses.

Additionally, we applied varies modern statistical learning techniques to the application of policyholder withdrawal behavior for the variable annuity product. The best statistical

learning method, which is selected according to 10-fold cross validation, performs better statistically compared to the traditional logistic regression method.

Finally, there are several directions for future research to be conducted. Our approximation heuristic uses the greedy algorithm by building the vine structure level by level, and future work remains to find a possibly better search heuristic that is able to look at future level trees when selecting the vine. Also, the optimality gap is computed via simulation data, and more research is needed to find a deterministic formula for the optimality gap. Lastly, the financial and policyholder withdrawal applications can be investigated further with better data sources and more statistical models.

Bibliography

Ahmed, N.A. and Gokhale, D.V. (1989). "Entropy Expressions and Their Estimators for Multivariate Distributions". Information Theory, IEEE Transactions on 35 (3): 688–692.

Anderson, T.W. (1958) An Introduction to Multivariate Statistical Analysis, *Wiley*, NewYork.

Arellano-Valle, R.B., Contreras-Reyes, J.E. and Genton, M.G. Shannon entropy and mutual information for multivariate skew-elliptical distributions. Scand. J. Stat. 2012, doi:10.1111/j.1467-9469.2011.00774.x.

Bedford T and Cooke R (2002) Vines--a new graphical model for dependent random variables. *Annuals of Statistics*, Volume 30, Number 4 (2002), 1031-1068.

Billor N, Hadi AS and Velleman PF (2000) BACON: blocked adaptive computationally efficient outlier nominators. *Computational Statistics & Data Analysis* 34, 279-298.

Briere-Giroux, G., Huet, J., Spaul, R., Staudt, A., Weinsier, D. "Predictive Modeling for Life Insurers", *Towers Watson*, 2010

DeMiguel V, Garlappi L and Uppal R (2009) Optimal versus naïve diversification: how inefficient is the 1/N portfolio strategy?. *Review of Financial Studies* 22:1915-53.

Diβmann, J., 2010. Statistical inference for regular vines and application. Diploma thesis, Technische Universitat Munchen.

Dissmann J, Brechmann E C, Czado C, Kurowicka D (2012) Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59, 52-69, 2013.

Donoho DL and Huber PJ (1983) The notion of breakdown point. *A Festschrift for Erich L. Lehmann*, P.J. Bickel, K.A. Doksum and J.L. Hodges, (eds.), 157-184, Belmont, CA: Wadsworth.

Friedman, N., Geiger, D. and Goldszmidt, M., "Bayesian Network Classifiers", *Machine Learning - Special issue on learning with probabilistic representations,* Volume 29 Issue 2-3, Nov./Dec. 1997, Pages 131 – 163

Garcia-Alvarez L and Luger R (2011) Dynamic correlations, estimation risk, and portfolio management during the financial crisis. Working Paper. CEMFI. Madrid.

Hardy, M. "Investment Guarantees: The New Science of Modeling and Risk Management for Equity-Linked Life Insurance", *John Wiley and Sons, Inc.,* Hoboken, New Jersey, 2003

Hastie, T., Tibshirani, R., Friedman, J. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", *Springer Series in Statistics*, 2009

Insured Retirement Institute, "Variable Annuity Sales Reach Pre-Crisis Levels", http://www.irionline.org/uploads/navaorg/news/608/original/q4_and_ye2011_sales_data.pdf, April 2012

Investopedia, "Chain-Weighted CPI", http://www.investopedia.com/terms/c/chain-linked-cpi.asp, July 2014

Investopedia, "Guaranteed Minimum Income Benefit - GMIB", http://www.investopedia.com/terms/g/gmib.asp, April 2012

Investopedia, "Guaranteed Minimum Withdrawal Benefit – GMWB", http://www.investopedia.com/terms/g/gmwb.asp, April 2012

Krueger, C. "Life and Annuity Products and Features", *Society of Actuaries Study Notes ILA-D105-07*, 2000

Kuncheva, L., Rodriguez, J. "An Experimental Study on Rotation Forest Ensembles", *MCS'07 Proceedings of the 7th international conference on Multiple classifier systems*, Pages 459-468, 2007

Kurowicka, D. and Cooke, R.M. (2006). Completion problem with partial correlation vines. Linear Algebra and Its Applications, 418(1):188–200.

Kurowicka D, Cooke RM and Callies U (2006), Vines Inference. *Brazilian Journal of Probability and Statistics*, 20, pp. 103-120.

Kurowicka D and Cooke RM (2006). Uncertainty Analysis with High Dimensional Dependence Modelling. *Wiley*, Chichester.

Kurowicka D and Joe H (2011). Dependence Modeling: Vine Copula Handbook. World Scientific Publishing Co. Pte. Ltd.

Maechler M and Stahel W (2009) Robust Scatter Estimators - The Barrow Wheel Benchmark. ICORS 2009, Parma.

Maronna RA, Martin DR and Yohai VJ (2006), Robust Statistics: Theory and Methods. *Wiley Series in Probability and Statistics*.

Maronna RA and Zamar RH (2002), Robust estimates of location and dispersion for high - dimensional data sets. *Technometrics*, 44:307 - 317.

Meier, L. "The group lasso for logistic regression", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* Volume 70, Issue 1, pages 53 – 71, February 2008

Morales-Napoles, O., 2008. Bayesian belief nets and vines in aviation safety and other applications. Ph.D. thesis, Technische Universiteit Delft.

Polsgrove, T. "Product Development Trends", *Society of Actuaries Study Notes ILA-D101-07*, 1999

Rodriguez, J., Kuncheva, L. "Rotation Forest: A New Classifier Ensemble Method", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Volume 28(10), Pages 1619 – 1630, 2006

Rousseeuw PJ and Leroy AM (1987), Robust Regression and Outlier Detection. *Wiley.*

Rousseeuw PJ and Driessen KV (1999), A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41, 212–223.

Rudin W (1976), Principles of Mathematical Analysis. *McGraw-Hill.*

Scotchie, R. "Incorporating Dynamic Policyholder Behavior Assumptions into Pricing of Variable Annuities", *Product Matters*, September 2006

Tyler DE (1987), A distribution-free M-estimator of multivariate scatter. *The Annals of Statistics*, 15, 234–251.

Williams V.V. (2011). "Breaking the Coppersmith-Winograd barrier", Unpublished manuscript

Witten, I., Frank, E., Hall, M. "Data Mining: Practical Machine Learning Tools and Techniques", *Elsevier*, Burlington, MA, 2011

Yule GU and Kendall MG (1965) An Introduction to the Theory of Statistics. 14th Ed, *Charles Griffin & Co.*, London