

# Value at Risk calculation using sparse vine copula models

Kwangjoon An<sup>a</sup>, Changryong Baek<sup>1,a</sup>

<sup>a</sup>Department of Statistics, Sungkyunkwan University

---

## Abstract

Value at Risk (VaR) is the most popular measure for market risk. In this paper, we consider the VaR estimation of portfolio consisting of a variety of assets based on multivariate copula model known as vine copula. In particular, sparse vine copula which penalizes too many parameters is considered. We show in the simulation study that sparsity indeed improves out-of-sample forecasting of VaR. Empirical analysis on 60 KOSPI stocks during the last 5 years also demonstrates that sparse vine copula outperforms regular copula model.

Keywords: value at risk, sparse vine copula, lasso, dependence structure

---

## 1. 서론

자산이 변동함에 따라 시장위험을 어떻게 관리할 것인가는 금융 시장의 가장 근본적인 관심사이다. 특히 2008년 서브프라임 모기지 사태, 2020년 코로나 사태 등 금융위기가 빈번하게 발생함에 따라 위험관리가 더욱 중요해지고 있다. 금융기관들이 위험관리수단으로 쓰이는 대표적인 지수는 최대예상손실액(Value at Risk, VaR)이다. VaR는 목표 시점에서 신뢰수준을 기준으로 나타낼 수 있는 최대 손실액으로 (Jorion, 2002), 하나의 숫자로 시장위험을 나타내기 때문에 직관적이고 간단하게 위험 정도를 나타낼 수 있는 장점이 있다. 본 논문은 최근 많은 관심을 받고 있는 구성종목이 다양한 포트폴리오, 고차원 다변량 금융자료에 대한 VaR 계산을 다루고 있다. 고차원 자료에서의 VaR를 정확하게 계산하기 위해서는 무엇보다 자산 간의 의존성, 특히 꼬리 의존성을 계산하는 것이 중요하다. 따라서 다변량 분포를 유동적으로 모델링할 수 있는 코플라 함수를 이용한 VaR 계산 방법을 연구하고자 한다.

코플라 함수는 일반적으로 금융자산의 수익률 결합분포를 모델링하는데, Embrechts 등 (2002)에서 밝혔듯이 코플라 함수는 자산 간의 의존성 구조(dependence structure)를 설명하는데 매우 유용하다. 하지만, 표준 코플라 함수는 의존성 구조를 하나의 코플라 함수에 의해 설명되기 때문에 포트폴리오의 구성종목이 다양한 고차원 자료의 모형에는 그 한계가 뚜렷하다. 예를 들어 타원형(elliptical) 코플라 함수인 다변량 정규(Gaussian) 혹은 스튜던트- $t$  (Student- $t$ ) 코플라는 꼬리 의존성이나 비대칭한 모델의 특징을 설명하기 적합하지 않다. 또한, 아르키메디안(Archimedean) 코플라는 비대칭한 특징을 일정 부분 설명하지만 하나 혹은 두 개의 모수만을 이용해 모델 적합을 하므로 다변량 모델링에 한계가 있다. 하나의 다변량 코플라 함수로 각 차원간 다른 성질을 가지는 다변량 분포를 모델링하기에는 부족하다. 이런 한계를 극복하기 위해 Bedford와 Cooke (2002)는 정규 바인 코플라(regular vine copula, R-vine) 함수를 제안했다. 바인 코플라는 이변량 코플라를 서로 연결하여 모델 구조를 형성하기 때문에 더 유연한 모델을 만들 수 있다는 장점이 있다. 하지만 모델 구조를

---

This work was supported by the Basic Science Research Program from the National Research Foundation of Korea (NRF-2019R1F1A1057104).

<sup>1</sup> Corresponding author: Department of Statistics, Sungkyunkwan University, 25-2, Sungkyunkwan-ro, Jongno-gu, Seoul, 03023, Republic of Korea. E-mail: crbaek@skku.edu

형성하는 이변량 코플라가 조건부 코플라일 때, 조건부 분포의 변수와 독립적이라는 가정때문에 최적해를 제공하는지 확신할 수 없고 계산 시간도 저하된다는 단점이 있다 (Müller와 Czado, 2019).

이러한 문제점을 해결하기 위해서 Müller와 Czado (2019)는 벌점화 방법인 lasso를 이용하여 성근(sparse) 의존성 구조를 선택하는 방법을 제시하였다. 하지만 Müller와 Czado (2019)이 제안한 성근 바인 코플라(sparse vine copula, L-vine) 방법은 금융자료의 VaR 추정에는 연구된 바가 없어 본 논문에서는 시뮬레이션과 실증 자료 분석을 통해 고차원 금융자료에서 성근 바인 코플라 모형이 VaR 추정에 있어서 얼마나 좋은 성능을 보여주는지 검증하고자 한다.

본 논문의 구성은 다음과 같다. 제2장에서는 방법론을 이해하기 위해 기초적인 배경설명을 하였고, 성근 바인 코플라를 사용하여 VaR를 구하는 과정에 대해서 요약하였다. 제3장에서는 성근 바인 코플라 구조의 성능에 대해서 시뮬레이션을 통해 알아본다. 제4장에서는 코스피 주가 지수를 바탕으로 VaR 추정에 대해서 실증 분석을 실시하였으며 마지막으로 제5장에서 본 논문의 결론과 논의점에 대해 서술하였다.

## 2. 모형 및 방법론

본 장에서는 논문에 쓰인 모형 및 추정 방법을 소개하고 기호(notation)를 정리하고자 한다.

### 2.1. 모형

시간  $t$ 에서의 자산 가격을  $P_t$ 라고 하자. 로그 수익률(log return)은 자산 가격의 변동으로 다음과 같이 정의된다.

$$r_t = \log(P_t) - \log(P_{t-1}).$$

본 논문에서는  $d$ 개의 자산이  $w_i \geq 0$ 의 가중치로 이루어진 포트폴리오

$$R_t := \sum_{i=1}^d w_i r_{t,i}, \quad \sum_{i=1}^d w_i = 1,$$

에 대한 최대예상손실액인 VaR의 계산에 목적을 두고 있다.

VaR는 포트폴리오의 손실에 대한 분포의 분위수를 뜻한다.  $100(1 - \alpha)\%$  VaR는,

$$P(R_t \leq VaR) = 1 - \alpha, \quad (2.1)$$

을 만족하는 실수값이다. 따라서 수익률 혹은 손실에 대한 분포를 알고 있다면 VaR는 단순히 분위수를 구하는 문제이다. 하지만 손실 분포를 구하기 쉽지 않기 때문에 다양한 방법으로 추정을 한다. 본 논문에서는 자산 수익률의 손실 분포를 구하기 위해 코플라 함수와 generalized autoregressive conditional heteroskedasticity (GARCH) 모형을 사용하여 모델에 적합하도록 하였다.

GARCH 모형은 이분산성을 가지는 금융자료를 분석하는 대표적인 모형으로 다음과 같이 정의된다.

$$r_{t,i} = \sigma_{t,i} \epsilon_{t,i}, \quad (2.2)$$

$$\sigma_{t,i}^2 = \omega_{0,i} + \sum_{j=1}^p \alpha_{j,i} r_{t-j,i}^2 + \sum_{k=1}^q \beta_{k,i} \sigma_{t-k,i}^2, \quad \omega_0 > 0, \alpha_{j,i} \geq 0, \beta_{k,i} \geq 0. \quad (2.3)$$

오차항  $\epsilon_{t,i}$ 는 평균이 0이고 분산이 1인 백색잡음이며 정상성을 가지기 위해서는  $\sum_{j=1}^p \alpha_{j,i} + \sum_{k=1}^q \beta_{k,i} < 1$ 을 만족해야한다. GARCH 모형의 적합에서는 차수  $p, q$ 의 선택이 중요하며 통상 AIC 혹은 BIC와 같은 정보통계량을 토대로 차수 선택이 이루어 진다. 본 논문에서는 GARCH(1, 1) 모형이 여러 복잡한 GARCH 확장 모형에 비교해서 결코 뒤쳐지지 않는다는 Park와 Baek (2014)의 연구 결과에 따라 GARCH(1, 1)을 편의상 사용하였고,

$t$ -분포를 사용하여 두터운 꼬리(heavy-tailedness)를 가지는 금융 자료의 특징을 반영하였다. 본 논문은 성근 바인 코플라 모형에 보다 중점을 두어 두터운 꼬리에 대한 모델링은 가장 기본적인  $t$ -분포를 사용하였다. 하지만, 두터운 꼬리를 좀 더 직접적으 모형에 반영시키는 Fan 등 (2014) 등의 방법을 사용한다면 모형을 확장하고 예측정확성을 높힐 수 있을 것이다.

## 2.2. 코플라(Copula) 함수

확률 변수  $\mathbf{X} = (x_1, \dots, x_d)$ 는  $d$ -차원의 다변량 확률 변수이며  $f(x_1, \dots, x_d)$ 는 결합확률밀도함수를  $F(x_1, \dots, x_d)$ 는 결합누적분포함수를 나타낸다. 주변누적분포함수(marginal cumulative density function)는  $F_i(x_i)$ 를 사용하였으며 주변확률밀도분포(marginal probability density function)는  $f_i(x_i)$ 라 표기하도록 하자. Sklar (1959)에 의해 정립된  $d$ -차원 코플라 함수  $C: [0, 1]^d \rightarrow [0, 1]$ 는 결합누적분포함수로 일변량 균등분포를 주변분포로 가진다. 코플라 함수는  $d$ -차원의 다변량 결합누적밀도함수를 주변누적분포함수  $F_1(x_1), \dots, F_d(x_d)$ 와 코플라  $C$ 를 바탕으로,

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)),$$

관계를 통해 모델링하는 기법이다. 만약  $F_i(\cdot)$ 가 연속이라면 코플라 함수가 유일하게 주어지며 결합누적분포함수가 절대연속형 함수이고 주변누적분포함수가 강한 증가함수 및 연속형 함수이면 코플라 밀도 함수  $c$ 는

$$c(F_1(x_1), \dots, F_d(x_d)) := \frac{\partial^d C(F_1(x_1), \dots, F_d(x_d))}{\partial F_1(x_1), \dots, \partial F_d(x_d)} = \frac{\partial^d F(x_1, \dots, x_d)}{\partial F_1(x_1), \dots, \partial F_d(x_d)} = \frac{f(x_1, \dots, x_d)}{\prod_{i=1}^d f_i(x_i)},$$

으로 나타낼 수 있으며,

$$f(x_1, \dots, x_d) = \prod_{i=1}^d f_i(x_i) c(F_1(x_1), \dots, F_d(x_d)), \quad (2.4)$$

으로 주어진다.

다변량 고차원 자료의 분포함수 추정에 있어서 변수간의 상호 의존성을 설명하기 위해서는 코플라 함수를 유연하게 선택할 수 있어야 한다. Joe (1996)는 이변량 코플라 함수를 서로 블록화를 이용해 연결하여 상호 의존성 문제를 해결하려고 했으며 이를 pair copula construction (PCC)라 부른다. 예를 들어  $d = 3$ 인 경우에,

$$f(x_1, x_2, x_3) = f_{3|1,2}(x_3|x_1, x_2) f_{2|1}(x_2|x_1) f_1(x_1),$$

에 대해서 수식 (2.4)를 사용하면,

$$f_{2|1}(x_2|x_1) = \frac{f_{12}(x_1, x_2)}{f_1(x_1)} = \frac{f_1(x_1) f_2(x_2) c_{12}(F_1(x_1), F_2(x_2))}{f_1(x_1)} = f_2(x_2) c_{12}(F_1(x_1), F_2(x_2))$$

으로 주어지고 비슷하게,

$$\begin{aligned} f_{3|1,2}(x_3|x_1, x_2) &= \frac{f_{13|2}(x_1, x_3|x_2)}{f_{12}(x_1|x_2)} = \frac{c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) f_{12}(x_1|x_2) f_{3|2}(x_3|x_2)}{f_{12}(x_1|x_2)} \\ &= c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) \times c_{23}(F_2(x_2), F_3(x_3)) f_3(x_3) \end{aligned}$$

으로 주어짐을 보일 수 있다. 따라서,

$$f(x_1, x_2, x_3) = f_1(x_1) f_2(x_2) f_3(x_3) \times c_{12}(F_1(x_1), F_2(x_2)) c_{23}(F_2(x_2), F_3(x_3)) \times c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2))$$

으로 분해됨을 알 수 있다. PCC는  $d$ -차원의 코플라 밀도함수를  $d(d-1)/2$ 개의 이변량(조건부) 코플라 밀도함수의 곱셈으로 나타내는 방법으로 각각의 이변량 코플라 밀도함수는 다른 코플라 함수에 얽매이지 않고 독립적이면서도 매우 유연하게 모델링을 할 수 있는 장점이 있다. 여기서 PCC를 통한 분해법은 유일하지 않다.

Bedford와 Cooke (2001)의 정규 바인(regular vine, R-vine)은 유일하지 않은 PCC의 해를 트리 구조로 정리한 방법으로 다음과 같다. 영문에 따른 표기법의 의미로써  $N$ 은 노드(nodes),  $E$ 은 모서리(edges) 그리고  $T$ 은 나무(tree)이다.  $T$ 은 그래프로써 연결되어 있는  $N$ 과  $E$ 로 구성된다.  $d$ -차원 정규 바인은  $d-1$ 개의  $N_i$ 와  $E_i$ 로 구성된 나무가 연결된 서열이고  $V = \{T_1, \dots, T_{d-1}\}$ 로 나타낸다. 첫번째 나무  $T_1$ 은 마디  $N_1 = \{T_1, \dots, T_{d-1}\}$ 과 모서리  $E_1$ 로 구성된다. 나무  $T_i$ ,  $i \geq 2$ 는 마디  $N_i = E_{i-1}$ 로 구성된다.  $\kappa_i(j)$ 는  $E_i$ 에서  $j$ 와 쌍을 이룬 변수를 구하는 할당(assignment) 함수이다.  $i = 2, \dots, d-1$ 이고  $\{a, b\} \in E_i$ 이면  $|a \cap b| = 1$ 을 만족하고 이를 근접조건이라 정의한다.

예를 들어,  $d = 4$ 인 경우에 위의 조건을 만족하는 다음과 같은 정규 바인 코플라 함수를 가질 수 있다.

$$C(u_1, u_2, u_3, u_4) = C_{12}(u_1, u_2) C_{23}(u_2, u_3) C_{34}(u_3, u_4) \\ \times C_{13|2}(u_{1|2}, u_{3|2}) C_{24|3}(u_{2|3}, u_{3|3}) \times C_{14|23}(u_{1|23}, u_{4|23}).$$

또한, 이러한 정규 바인은 구조행렬(structure matrix)  $R$ 로 간단하게 나타낼 수 있다.

$$R = \begin{pmatrix} 4 & & & \\ 1 & 3 & & \\ 2 & 1 & 2 & \\ 3 & 2 & 1 & 1 \end{pmatrix}. \quad (2.5)$$

해석하는 방법은  $T_1$ 에서  $N_1 = \{1, 2, 3, 4\}$ 으로 노드를 구성하고 있고, 대각원소의 근접한 짝을 통해  $E_1 = \{(4, 3), (3, 2), (2, 1)\}$ 임을 알 수 있다. 정규 바인의 규칙에 따라  $T_2$ 에서  $N_2 := E_1$ 이다. 추가로 조건부 구조를 설명하기 위해서는 가장 아래 행부터 조건부로 해석을 하면 된다. 첫째 열에서 노드 (3)이 조건부로 주어진 경우에 연결된 모서리는 (4, 2)이고, 두번째 열에서 노드 (2)가 주어진 경우에는 (3, 1)이 연결되어 있다. 이를 통해  $N_2 = \{(4, 3), (3, 2), (2, 1)\}$ ,  $E_2 = \{(1, 3|2), (2, 4|3)\}$ 임을 알 수 있다.  $T_3$ 의 구조를 알기 위해서는 아래 두 행을 고정 시킨 뒤 해석하면 된다. 첫번째 열에서 (2, 3)이 주어진 경우에 연결된 노드는 (4, 1)이므로  $N_3 := E_2 = \{(1, 3|2), (2, 4|3)\}$ ,  $E_3 = \{(4, 1|2, 3)\}$ 임을 알 수 있다. 따라서 구조행렬을 통해서 정규 바인 코플라의 구조를 알 수 있으며, 같은 구조라 할지라도  $R$ 이 유일하게 표현되지 않는다.

### 2.3. 성근 바인(L-vine) 코플라 모형

정규 바인 코플라는 각각의 이변량 분포에 다른 코플라 함수를 적용할 수 있다는 장점이 있다. 바인 코플라는  $d(d-1)/2$ 개의 쌍을 바탕으로 종속성을 표현함에 있어 유용하며 정규 바인 구조를 통해서 나타낼 수 있다. 정규 바인 구조는 Dissmann 등 (2013)의 알고리즘을 흔히 사용한다. 하지만, 차원이 작을 경우 최적해를 잘 찾아주지만 차원이 커질수록 최적해를 적절히 제공하지 못하는 문제점이 있다 (Müller와 Czado, 2019). 예를 들어, 정규 바인 구조를 이용한 바인 코플라에서는 차원이 20보다 큰 경우 계산이 너무 방대해짐에 따라 계산 시간이 증가하고 최적해를 구하기 어렵다. 그 결과 고차원에서 계산을 단순하게 함으로써 시간을 줄이고 적절한 구조를 찾아 최적해를 구하기 위한 연구가 진행되었고 Müller와 Czado (2019)는 lasso를 이용하여 정규 바인 구조를 선택하는 방법을 제안하였다. Müller와 Czado (2019)는 성근 바인(L-vine) 코플라 구조를 찾기 위해서 구조방정식 모델(structure equation model)을 바탕으로 lasso를 대입한다. 먼저, 순서(ordering) 함수  $\eta(i)$ 가 각각의  $i = 1, \dots, d$ 에 대해 존재해야 하고 2.2절에서 설명한 할당 함수  $\kappa_i$ 가 존재해야 한다. 그다음,  $X_i$

와 0이 아닌 계수를 가지는 변수들의 집합  $R_1(j)$ ,  $j = 1, \dots, d$  을 바탕으로 구조방정식을 만들어 튜닝 모수  $\lambda$ 와 함께 lasso를 적용한다. 본 절에서는 예시를 통해 방법론을 간략히 설명하고자 한다. 자세한 내용은 Müller와 Czado (2019)을 참고하기 바란다.

예를 들어 수식 (2.5)의 순서 함수는  $\eta(j) = j$ 로 정의할 수 있다. 순서 함수는 독립변수로 빈번하게 사용되는 변수에는 작은 값을 주고 적게 사용되는 변수에 큰 값을 준다. 앞에서의 예시를 따르면  $X_1, X_2, X_3, X_4$ 를 다음과 같은 구조방정식 모델로 나타낼 수 있다.

$$\begin{aligned} X_1 &= \psi_1 \epsilon_1, \\ X_2 &= \beta_{2,1} X_1 + \psi_2 \epsilon_2, \\ X_3 &= \beta_{3,1} X_1 + \beta_{3,2} X_2 + \psi_3 \epsilon_3, \\ X_4 &= \beta_{4,1} X_1 + \beta_{4,2} X_2 + \beta_{4,3} X_3 + \psi_4 \epsilon_4. \end{aligned}$$

이를 통해 얻어지는 구조방정식을 바탕으로 설명변수  $j$ 와 관계된 변수들을 할당 함수  $\kappa_j$ 라 한다. 수식 (2.5)의 할당함수는  $k_1(4) = 3, k_2(4) = 2, k_3(4) = 1$  이다. 할당 함수의 해석은 예를들어  $E_3 = \{(4, 1|2, 3)\}$ 이므로  $E_3$  에서 4와 관계를 맺은 변수는 1이다.

성근 구조행렬  $M$ 에 대해서  $(m_{1,1}, \dots, m_{d,d})$ 을 대각원소라고 가정하자. 예를 들어,  $(4, 3, 2, 1)$ 를 대각원소로 가지는  $4 * 4$  행렬을  $M$ 이라 하자.  $m_{4,3} = m_{4,4} = 1$ 이고 나머지  $j = 3, 4$ 에 대한 열에 대하여 (2.6) 수식을 통해 각각의 대각원소와  $T_1$ 에서 모서리를 형성하는 변수를 구한다. 다음과 같은 과정이 진행된 후  $T_1$ 이 정해지면 행렬  $M$ 은 다음과 같이 나타낼 수 있으며  $T_1$ 에서  $(j, k_j)$ ,  $j = 2, \dots, d$  에 해당하는 모서리를 가지고 있다. 이 방법을 다음 단계의 나무에 적용할 수 있으나 다음 나무에서는 근접 조건을 만족하는지 유의해야 한다.

$$M := \begin{pmatrix} m_{1,1} & & & \\ m_{2,1} & m_{2,2} & & \\ m_{3,1} & m_{3,2} & m_{3,3} & \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} \end{pmatrix} = \begin{pmatrix} 4 & & & \\ & 3 & & \\ & & 2 & \\ \kappa_1(4) & \kappa_1(3) & 1 & 1 \end{pmatrix}.$$

예를 들어,  $T_1$ 에서 다음과 같이 정해져서 행렬  $M$ 이 다음과 같이 정해졌다면  $T_2$ 에서  $m_{3,1} = 2$  만 가능하다.

$$M = \begin{pmatrix} 4 & & & \\ & 3 & & \\ \square & 1 & 2 & \\ 3 & 2 & 1 & 1 \end{pmatrix}.$$

만약  $m_{3,1}$ 이 정칙경로에 의해  $m_{3,1} = 1$ 이 된다면  $T_1$ 에서 (1,3)을 포함한 모서리가 없어 근접 조건을 만족하지 못하여  $m_{3,1} = 2$  만 가능하고 이를 고려하여 lasso를 적용하여야 한다.

첫 번째 나무에서는 근접 조건에 대해 고려할 필요가 없었지만 두 번째 나무부터는 근접 조건을 고려해야 한다. 정칙경로(regularization path)에 의해 변수가 정해진다 하더라도 근접 조건이 우선시 되어 조건을 만족하는 변수를 정하여야 하고 이를 포함한 정칙경로를 계산하여야 한다. 이때 계산은 페널티가 없는 항에는 이전 나무에서 설명변수와 모서리를 형성한 변수를 포함하고, 페널티가 있는 항에는 각각의 나무에서 설명변수와 모서리를 형성한 변수만을 포함해야 한다. 이를 lasso 문제로 표현하면 다음과 같다.

$$\underset{\varphi}{\text{minimize}} \left( \frac{1}{2n} \sum_{i=1}^n \left( x_{i,j} - \sum_{\ell \in \mathcal{H} \setminus \mathcal{B}} \varphi_{j,\ell} x_{i,\ell} \right)^2 + \sum_{\ell \in \mathcal{H} \setminus \mathcal{B} \setminus \mathcal{W}} \lambda_{j,\ell} |\varphi_{j,\ell}| \right). \tag{2.6}$$

수식 (2.6)에서  $\mathcal{H}$ 는  $X_j$ 와 블록화된 모든 변수의 집합,  $\mathcal{B}$ 는 계산식에 해당하는 나무 이후에  $X_j$ 와 블록화된 집합,  $\mathcal{W}$ 는 계산식 이전의 나무에서  $X_j$ 와 블록화된 집합을 의미한다.

수식 (2.6)을 토대로, 두 번째 나무에서  $X_4$ 의 정칙경로는 수식 (2.7)과 같이 나타낼 수 있다.  $X_4$ 와 블록화된 모든 변수는 집합  $\mathcal{H}(3, 1) = \{3, 2, 1\}$ 이다.  $X_4$ 의 독립변수로 세 번째 나무에서 블록화된  $X_1$ 은 남은 집합  $\mathcal{B}$ 가 되고  $\mathcal{B}(3, 1) = \{1\}$ 이다.  $\mathcal{B}$ 가 정해지면 다시 정칙경로를 계산하는데, 이때 계산은 페널티가 없는 항에 첫 번째 나무  $T_1$ 에서  $X_4$ 와 블록화된  $X_3$ 은 포함되고, 페널티가 있는 항에는 첫 번째 나무  $T_1$ 에서 연결된  $X_3$ 은 연결된 집합  $\mathcal{W}(3, 1) = \{3\}$ 이다. 따라서  $T_2$ 에서  $X_4$ 에 대한 정칙경로는 다음과 같이 정의한다.

$$\underset{\varphi}{\text{minimize}} \left( \frac{1}{2n} \sum_{i=1}^n \left( x_{i,4} - \sum_{\ell \in (3,2,1) \setminus (1)} \varphi_{4,\ell} x_{i,\ell} \right)^2 + \sum_{\ell \in (3,2,1) \setminus (1) \setminus (3)} \lambda_{4,\ell} |\varphi_{4,\ell}| \right). \quad (2.7)$$

성근 바인 코플라 모형의 추정에는 Friedman 등 (2010)의 glmnet R-패키지를 통해 구현될 수 있다. Lasso를 이용한 성근 바인 코플라 방법은 구조방정식을 바탕으로 해를 찾기에 정규 바인보다 고차원에서 더 향상된 속도와 최적해를 구할 수 있다. 성근의 정도는 페널티 튜닝 모수인  $\lambda$ 를 통해 결정되는데 lasso에서 많이 쓰이는 상호교차법이나 정보통계량을 이용하는 방법을 쓸 수 있다.

종합하여 VaR 추정은 다음의 방법을 통해서 이루어진다. Brechmann과 Czado (2013)의 방법과 유사하게 진행했다. 또한 정규 분포를 사용한 코플라 모형은 VaR 추정에 있어서 적합하지 않다는 MacKenzie와 Spears (2014)의 연구를 바탕으로, 정규분포보다는 다른 분포를 사용하는 것이 권장되며 본 연구에서는  $t$ 분포를 사용하여 두터운 꼬리를 가지는 금융자료의 특징을 반영하도록 하였다.

1. 수익률 자료를 GARCH 모델을 이용해 추정하고 이를 통해 변동성  $\hat{\sigma}_t (t = 1, \dots, n)$ 를 구하여 표준화 잔차  $\hat{\epsilon}_t = r_t / \hat{\sigma}_t$ 를 얻는다.
2. 1번 과정에서 구한 표준화 잔차를 누적분포함수(예를 들어,  $t$  분포)에 대입해 0과 1 사이의 코플라 데이터  $\hat{F}_i(x) = P(\hat{\epsilon}_i \leq x)$ 로 만든다.
3. 코플라 데이터를 바인 코플라 모델에 적합한다. 추정된 코플라 모델을 통해,  $m = 1, \dots, M$ 개의 시뮬레이션 벡터를 생성해 표준화 잔차  $\hat{\epsilon}_{n+1}^m = (\hat{\epsilon}_{n+1,1}^m, \dots, \hat{\epsilon}_{n+1,d}^m)$ 를 예측한다. 각각의 벡터를 통해  $R_{n+1}^m \approx \sum_{i=1}^d w_i \hat{r}_{n+1,i}^m$  이때  $\hat{r}_{n+1,i}^m = \hat{\sigma}_{n+1,i} \hat{\epsilon}_{n+1,i}^m$ 이다.
4. 마지막으로 VaR은  $\widehat{\text{VaR}}_{n+1}^\alpha = \{q : M^{-1} \sum_{m=1}^M I(R_{n+1}^m \leq q) = \alpha\}$ 로 추정한다. 여기서  $I(\cdot)$ 는 지시 함수다.

### 3. 시뮬레이션 연구

본 장에서는 정규 바인 구조와 성근 바인 구조를 바탕으로 각각 계산한 VaR을 비교함으로써 성능을 비교하고자 한다. 성능 비교의 지표로는 표본 외 예측(out-of-sample forecasting)을 사용하여 VaR에 대한 경험 empirical mean square prediction error (MSPE),

$$\text{MSPE} = \mathbb{E} \left( \widehat{\text{VaR}}(R_t) - \text{VaR}(R_t) \right)^2, \quad (3.1)$$

를 통해 비교하고자 한다. 경험적 평균은 100번의 반복을 통해서 얻었다. 모의 실험에서 사용한 모형은 실제 금융 자료의 특징을 그대로 반영하고자  $d$ -차원의 코스피 종목 및 기간을 무작위로 골라 단변량 GARCH(1, 1) 모형을 적합하여 모형 계수를 생성하였다. 또한 GARCH(1, 1) 모형을 적합한 뒤 얻은 잔차들에 대해서 다변량 정규분포를 사용하여 적합하여 이를 바탕으로 모형을 생성하였다. 또한 성근 모형을 만들기 위해서 다변량 정규분포의 모수인  $\Sigma$  행렬에서 비대각(off-diagonal) 원소가 0.2보다 작은 경우 이를 0으로 만들었다. 차원  $d = 10$ 인 경우 금호석유, 미래에셋증권, 이마트, NAVER, GS리테일, SK이노베이션, 대우건설, 신세계, 씨에스윈드, 한화생명의 계수를 사용했고,  $d = 30$ 인 경우 카카오, 삼성전자, CJ대한통운, NH투자증권, 일진머티리얼즈,

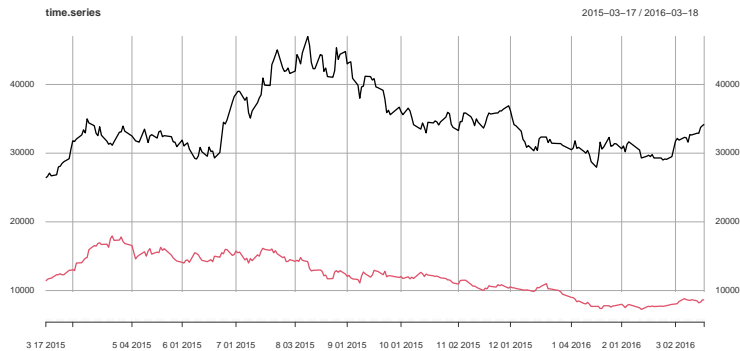


Figure 1: Time plot of two assets.

한진칼, LG디스플레이, 동서, 셀트리온, 기업은행, 신한지주, 고려아연, 신풍제약, SK텔레콤, KB금융, 현대제철, POSCO, 삼성물산, 아모레G, SK를 추가했다. 또  $d = 50$ 인 경우 에스원, 만도, 삼성엔지니어링, 삼성전기, S-OIL, 강원랜드, SK하이닉스, GS, LG전자, 현대글로벌비스, 삼성화재, 현대건설, 현대모비스, 한국타이어앤테크놀로지, 두산중공업, 코웨이, LG이노텍, 현대미포조선, 키움증권, 롯데쇼핑을 추가했다.

Figure 1은 동서와 미래에셋증권과의 관계를 나타낸 그림이다. Figure 1의 맨 윗 그림은 2015년 3월 17일부터 2016년 3월 18일까지 가격을 나타낸 그래프이다. 자기 상관함수 그림 및 교차상관함수를 통해 두 자산 간의 무시할 수 없는 상관관계가 존재함을 확인하였으며 이는 다변량 코플라 모형이 적절함을 보여준다. 또한 GARCH 모형이 타당한지 살펴보기 위해서 Engle의  $F$ -테스트를 ARCH(3)를 사용하여 동서의 로그 수익률을 적용한 결과 카이제곱 값이 32.14,  $p$ -값이  $4.89 \times 10^{-7}$ 로 ARCH effect가 있음을 확인할 수 있었다.

시뮬레이션 순서는 다음과 같다. 포트폴리오의 가중치  $w_i = 1/d$ 로 동일한 가중치를 사용하였다.

- Step1 주어진 차원  $d$  및 표본수  $n$ 에 대해서 앞서 설명한 GARCH(1, 1) 및 다변량 정규분포를 바탕으로  $d$ -차원의 확률변수를  $n + 50$ 개 생성한다. 처음  $n$ 개의 자료는 트레인 셋(train set)이며 마지막 50개의 자료를 테스트 셋(test set)이다. 참값인  $\text{VaR}(R_t)$ ,  $t = n + 1, \dots, n + 50$  역시 100번의 표본추출을 통해서 구한다.
- Step2 트레인 셋에 대해서 일변량 GARCH(1, 1) 모형에 열 별로 적합한다. 각각의 표준화 잔차를  $t$ -분포에 대입하여 0과 1 사이의 데이터로 만든다.
- Step3 Step2에서 얻어진 트레인 셋에 대해서 각각 정규 바인 구조(R-vine)와 성근 바인 구조(L-vine)에 적합하여 잔차를 추정하고, GARCH(1, 1) 모형을 토대로 예측한  $\hat{\sigma}_t$ 를 통해  $R_t$ 를 계산한다.
- Step4 Step3의 모형에 기반하여 1-step-ahead 표본 외 예측값을 구하였고, 이를 바탕으로 추정된 90%, 95%, 99% VaR을 1번 단계에서 구한 참값  $\text{VaR}(R_t)$ 와 비교하여 경험적 MSPE를 구하였다.

실제 모형 적합에 쓰인 R-패키지는 다음과 같다. R-패키지인 rugarch의 ugarchfit 함수를 사용해 단변량 GARCH(1, 1)을 적합하였고, ugarchroll 함수를 사용해 테스트 셋에서 롤링 윈도우(rolling window)를 사용해 1-step-ahead  $\sigma_t$ 를 예측했다. GARCH(1, 1) 모형으로 구해진 표준화된 잔차를 코플라 함수에 적용하기 위해서는 MASS R-패키지의 ftdistr 함수를 이용해 자유도를 찾아  $t$ -분포에 적용하여 0과 1 사이의 값으로 만들었다. 정규 바인 코플라 모형은 VineCopula R-패키지를 사용했다 (Nagler 등, 2021). 성근 바인 코플라 함수는 Müller와 Czado (2019)에 소개되어 있는 <https://github.com/dtmueller/VineC-opulaHD>의 패키지를 이용해 적합하였다.

시뮬레이션 결과는 Table 1으로 나타내었다. 정규 바인 코플라와 성근 바인 코플라의 MSPE를 99%, 95%, 90% VaR을 나누어 비교했다. 99% VaR이 가장 극단의 값을 추정하는 점을 고려했을 때, 95%, 90% VaR과

Table 1: Result of simulation

|           |        |        | $d = 10$ | $d = 30$ | $d = 50$ |
|-----------|--------|--------|----------|----------|----------|
| $n = 250$ | R-vine | 90%VaR | 3.32     | 1.33     | 0.79     |
|           |        | 95%VaR | 5.50     | 1.85     | 1.30     |
|           |        | 99%VaR | 12.99    | 5.07     | 3.77     |
|           | L-vine | 90%VaR | 3.30     | 1.26     | 0.65     |
|           |        | 95%VaR | 5.47     | 1.81     | 1.10     |
|           |        | 99%VaR | 12.74    | 5.02     | 3.41     |
| $n = 450$ | R-vine | 90%VaR | 2.71     | 1.08     | 0.71     |
|           |        | 95%VaR | 4.66     | 1.90     | 1.14     |
|           |        | 99%VaR | 10.88    | 4.95     | 3.29     |
|           | L-vine | 90%VaR | 2.75     | 1.01     | 0.55     |
|           |        | 95%VaR | 4.81     | 1.77     | 0.90     |
|           |        | 99%VaR | 11.62    | 4.81     | 2.63     |
| $n = 950$ | R-vine | 90%VaR | 2.48     | 0.93     | 0.74     |
|           |        | 95%VaR | 3.99     | 1.36     | 1.12     |
|           |        | 99%VaR | 8.84     | 3.52     | 2.30     |
|           | L-vine | 90%VaR | 2.48     | 0.85     | 0.60     |
|           |        | 95%VaR | 3.95     | 1.30     | 0.85     |
|           |        | 99%VaR | 8.53     | 3.27     | 1.77     |

비교하여 가장 큰 MSPE를 가지는 것이라 판단된다. 차원은 고정하고 데이터의 크기가 커질수록 MSPE가 점점 줄어들음을 알 수 있고, 데이터 크기를 고정하고 차원이 커질 수록 MSPE가 마찬가지로 줄어들음을 알 수 있다 이는 포트폴리오 구성에서 분산투자의 효과가 점점 커진 것으로 볼 수 있다. 차원이 10인 경우, 정규 바인 코풀라와 성근 바인 코풀라의 결과값이 서로 큰 차이가 나지 않았다. 하지만 차원이 점차 커짐에 따라 성근 바인 구조의 결과값이 정규 바인 구조의 결과값보다 향상되었음을 알 수 있다. 특히 차원이 50이고, 데이터의 개수가 250인 경우 성근 바인 구조의 결과값과 정규 바인 구조의 결과값이 큰 차이를 보여 고차원 및 표본의 개수가 작은 경우에 성근 모형의 성능이 더 좋음을 확인할 수 있었다. 이는 성근 모형의 핵심 방법인 lasso 추정에서 널리 관측되는 바로, 추정하는 모수의 개수를 줄임으로 전체 오차를 줄이는 데에서 기인한 것이라 생각할 수 있다.

#### 4. 실증 자료 분석

본 장에서는 정규 바인 구조와 성근 바인 구조의 성능을 실증 자료를 바탕으로 비교해보고자 한다. 먼저 TIGER TOP10 ETF의 보유종목인 삼성전자, SK하이닉스, NAVER, 카카오, 삼성SDI, LG화학, 현대차, 셀트리온, 기아, POSCO 10개의 종목에서 두 구조의 성능을 비교해보고, 추가로 TIGER TOP30 ETF에 추가로 코스피에서 시가총액 및 업종을 고려하여 현대모비스, LG생활건강, SK, 엔씨소프트, HMM, 한국전력, LG, 삼성에스디에스, 하나금융지주, 포스코케미칼, 두산중공업, KT&G, S-Oil, 대한항공, 고려아연, CJ제일제당, KT, 한온시스템, 한화솔루션, 금호석유, LG유플러스, 코웨이, 한국금융지주, 한국타이어엔테크놀로지, SKC, 강원랜드, LG이노텍, 아모레G, 맥쿼리인프라, 한진칼을 더해 총 60개의 종목으로 포트폴리오를 구성하였다. 2015년 1월 1일부터 2018년 12월 31일까지의 총 975일의 수익률을 바탕으로 2019년 1월 02일부터 12월 30일까지의 총 254일의 VaR을 예측하고 결과를 비교했다.

데이터는 Yahoo.finance에서 각각의 종목에 대한 일별 증가를 Quantmod R-패키지를 이용하여 얻었다



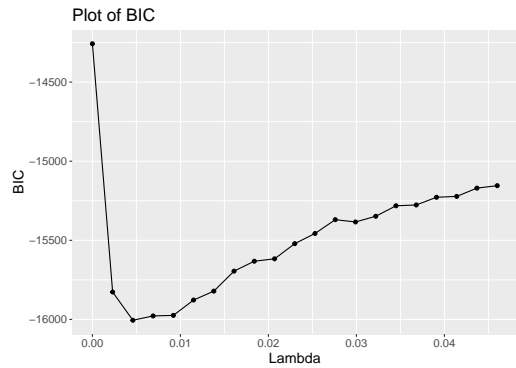
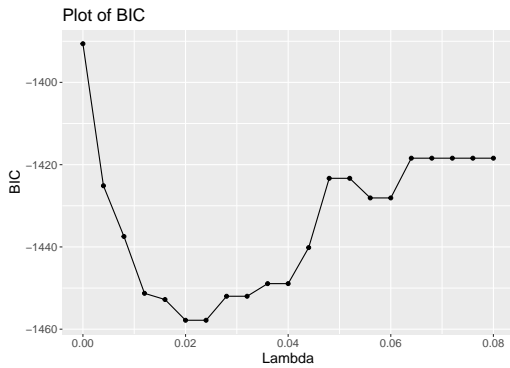


Figure 2: BIC plot for  $d = 10$ (left) and BIC plot for  $d = 60$ (right).

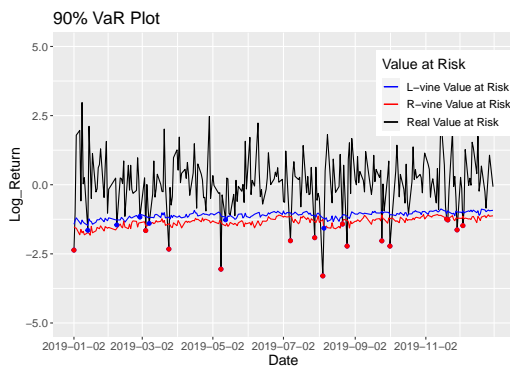


Figure 3: 90% VaR plot for  $d = 10$ .

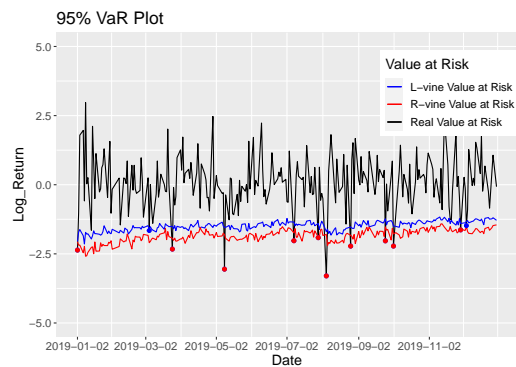


Figure 4: 95% VaR plot for  $d = 10$ .

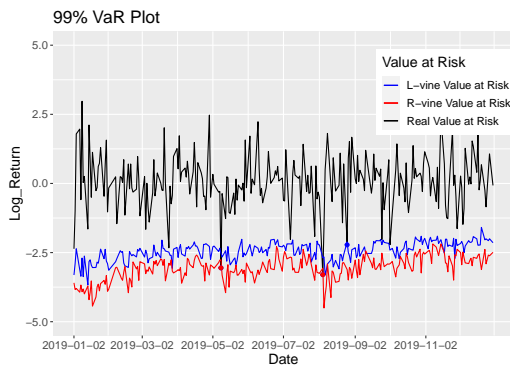


Figure 5: 99% VaR plot for  $d = 10$ .

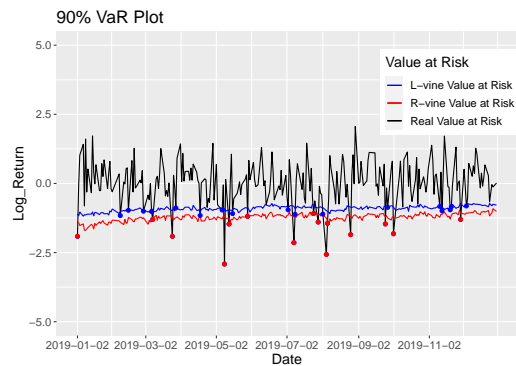


Figure 6: 90% VaR plot for  $d = 60$ .

(Ryan과 Ulrich, 2020). VaR의 예측은 2.3절 및 시뮬레이션 연구에서 소개한 알고리즘을 토대로 계산하였다. 다만, 성근 바인 구조에서 페널티 튜닝 모수인  $\lambda$ 를 선택할 때는 자료에 의존(data adaptive)한 방법인 BIC를 기준으로 정했다. Figure 2에서 살펴보듯이  $d = 10$ ,  $d = 60$ 인 경우 각각 튜닝 모수  $\lambda = 0.02$ ,  $\lambda = 0.0046$ 에서 가장 작은 BIC값을 가짐을 확인할 수 있다.

차원이 10인 경우 R-vine과 L-vine의 결과값을 각각 90%, 95%, 99% VaR에 따라 Figures 3, 4, 5로 나타

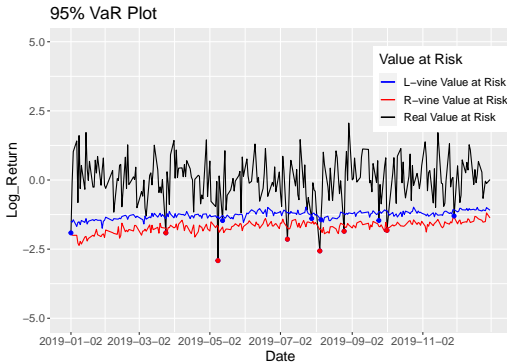


Figure 7: 95% VaR plot for  $d = 60$ .

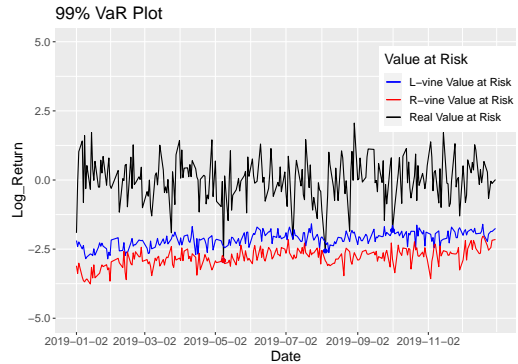


Figure 8: 99% VaR plot for  $d = 60$ .

Table 2: Kupiec test when  $d = 10$

| $VaR_{\alpha}$               | Expected exceedances | Actual exceedances | H0 "Correct exceedances" |
|------------------------------|----------------------|--------------------|--------------------------|
| Kupiec test for R-vine model |                      |                    |                          |
| $VaR_{0.1}$                  | 24                   | 15                 | $p=0.0304$               |
| $VaR_{0.05}$                 | 12                   | 10                 | $p=0.4963$               |
| $VaR_{0.01}$                 | 2                    | 2                  | $p=0.7654$               |
| Kupiec Test for L-vine model |                      |                    |                          |
| $VaR_{0.1}$                  | 24                   | 21                 | $p=0.4459$               |
| $VaR_{0.05}$                 | 12                   | 12                 | $p=0.9414$               |
| $VaR_{0.01}$                 | 2                    | 3                  | $p=0.7330$               |

Table 3: Kupiec test when  $d = 60$

| $VaR_{\alpha}$               | Expected exceedances | Actual exceedances | H0 "Correct exceedances" |
|------------------------------|----------------------|--------------------|--------------------------|
| Kupiec test for R-vine model |                      |                    |                          |
| $VaR_{0.1}$                  | 24                   | 15                 | $p=0.0304$               |
| $VaR_{0.05}$                 | 12                   | 6                  | $p=0.0429$               |
| $VaR_{0.01}$                 | 2                    | 1                  | $p=0.2907$               |
| Kupiec test for L-vine model |                      |                    |                          |
| $VaR_{0.1}$                  | 24                   | 32                 | $p=0.1253$               |
| $VaR_{0.05}$                 | 12                   | 12                 | $p=0.9414$               |
| $VaR_{0.01}$                 | 2                    | 3                  | $p=0.7330$               |

내었다. 차원이 60인 경우 R-vine과 L-vine의 결과값을 각각 90%, 95%, 99% VaR에 따라 Figures 6, 7, 8로 나타내었다. 그래프에서 검은 선은 실제 VaR을, 빨간 선은 R-vine의 VaR 추정값을, 파란 선은 L-vine의 VaR 추정값을 나타낸다. 빨간 점은 실제 수익률 중 추정된 R-vine의 VaR보다 큰 손실률을, 파란 점은 실제 수익률 중 추정된 L-vine의 VaR보다 큰 손실률을 나타낸다. Müller와 Czado (2019)에서 L-vine의 계산시간이 항상 되었다는 결과를 바탕으로, tictoc R-패키지를 통해 차원이 10인 경우 R-vine의 경우 277초, L-vine의 경우 14초, 차원이 60인 경우 R-vine의 경우 177분, L-vine의 경우 14분이 소요되어 성근 바인 구조의 계산 속도가 향상되었음을 확인할 수 있었다. 실제 데이터의 VaR의 실패율이 목표하는 VaR의 신뢰수준과 비교하는 Kupiec (1995)에서 제안한 Kupiec 검정을 통해 사후검정(backtest)를 실시하였다. Kupiec 검정에서의 귀무가설은 ‘정

확한 예측을 했다'이다. 차원이 10인 경우의 정규 바인 코플라 모형과 성근 바인 코플라 모형을 이용한 Kupiec 사후검정 결과는 Table 2, 차원이 60인 경우의 정규 바인 코플라 모형과 성근 바인 코플라 모형을 이용한 Kupiec 사후검정 결과는 Table 3에 정리하였다.

Tables 2–3에서 expected exceedances는 기대 초과량이고, actual exceedances는 분석을 바탕으로 한 실제 초과량이다.  $H_0$  “Correct exceedances” 항에는 결과값에 대한 유의 확률을 기입했다. 차원이 10인 경우, 90% VaR는 성근 바인 모형을 이용한 예측이 24개 중 21개를 예측했으므로 정규 바인 모형을 이용한 예측보다 우월한 반면, 95%와 99% VaR은 정규 바인 모형을 이용한 예측이 각각 12개 중 11개, 2개 중 2개를 예측함으로써 더 우월한 결과를 보였다. 차원이 60으로 증가한 경우, 정규 바인 모형은 90%, 95%에서 Kupiec 점정 결과 보편적인 유의 확률 0.05보다 작으므로 귀무가설을 기각한다. 반면 성근 바인 모형은 90% VaR에서 24개 중 32개, 95% VaR에서 12개 중 12개, 99% VaR에서 2개 중 3개를 예측했다. 따라서 차원이 작은 경우 정규 바인 모형과 성근 바인 모형의 성능의 성능 차이는 크지 않으나, 차원이 점차 커짐에 따라 성근 바인 모형의 성능이 향상됨을 확인할 수 있었다.

## 5. 결론 및 논의점

본 논문에서는 포트폴리오의 구성이 다양해지고 고차원 데이터에서의 위험관리가 중요해짐에 따라 코플라 함수를 이용해 포트폴리오의 VaR 추정을 다루었다. 구성 자산들 간의 의존성 구조를 효과적으로 모델링할 수 있는 정규 바인 코플라를 바탕으로 고차원 포트폴리오에서의 추정오차를 줄이기 위해서 성근 바인 코플라를 사용하여 VaR 예측력을 비교하였다. 시뮬레이션 연구에서 차원이 증가할수록 성근 바인 코플라 모형이 더 적은 예측 오차를 줄을 확인할 수 있었다. 또한 코스피 자료를 활용한 실증 자료 분석에서도 비슷한 결과를 얻었는데 차원이 10인 경우, 정규 바인 구조와 성근 바인 구조 큰 차이를 보이지 않았다. 하지만 차원이 60으로 커진 경우, 성근 바인 구조가 정규 바인 구조보다 90%, 95%, 99% VaR 모두 향상된 예측 결과를 보여 주었다. 이는 고차원 금융자료의 VaR추정에 있어서 성근 바인 모형이 정규 바인 모형보다 더 향상된 성능을 보여준다고 볼 수 있겠다. 한편, 성근 바인 모형은 별점화 튜닝 모수  $\lambda$ 에 의해 많은 영향을 받는다. 그러므로 정보통계량을 비롯 자료에 기반한  $\lambda$ 를 추정하는 것이 예측성능을 크게 좌우하므로 이를 어떻게 잘 선택할 것인가에 대한 추가 연구도 필요하다. 또한 elastic net, SCAD 등의 다른 별점화 방법을 통해서 성근 바인 코플라 모형을 적용할 경우에 성능의 향상을 기대할 수 있는지에 대해서도 흥미로운 후속 연구가 될 것이다.

## References

- Bedford T and Cooke RM (2001). Probability density decomposition for conditionally dependent random variables modeled by vines, *Annals of Mathematics and Artificial intelligence*, **32**, 245–268.
- Bedford T and Cooke RM (2002). Vines: A new graphical model for dependent random variables, *Annals of Statistics*, **30**, 1031–1068.
- Brechmann EC and Czado C (2013). Risk management with high-dimensional vine copulas: An analysis of the Euro Stoxx 50, *Statistics & Risk Modeling*, **30**, 307–342.
- Dissmann J, Brechmann EC, Czado C, and Kurowicka D (2013). Selecting and estimating regular vine copulae and application to financial returns, *Computational Statistics and Data Analysis*, **59**, 52–69.
- Embrechts P, McNeil A, and Straumann D (2002). Correlation and dependence in risk management: properties and pitfalls, *Risk Management: Value at Risk and Beyond*, **1**, 176–223.
- Fan J, Qi L, and Xiu D (2014). Quasi-maximum likelihood estimation of GARCH models with heavy-tailed likelihoods, *Journal of Business & Economic Statistics*, **32**, 178–191.

- Friedman J, Hastie T, and Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent, *Journal of Statistical Software*, **33**, 1–22.
- Joe H (1996). Families of  $m$ -variate distributions with given margins and  $m(m - 1)/2$  bivariate dependence parameters, *Lecture Notes-Monograph Series*, 120–141.
- Jorion P (2002). *Value at Risk: The New Benchmark for Managing Financial Risk*, McGraw-Hill.
- Kupiec P (1995). Techniques for verifying the accuracy of risk measurement models, *The Journal of Derivatives*, **3**, 73–84.
- MacKenzie D and Spears T (2014). ‘A device for being able to book P&L’: The organizational embedding of the Gaussian copula, *Social Studies of Science*, **44**, 418–440.
- Müller D and Czado C (2019). Selection of sparse vine copulas in high dimensions with the lasso, *Statistics and Computing*, **29**, 269–287.
- Nagler T, Schepsmeier U, Stoeber J, Brechmann EC, Graeler B, and Erhardt T (2021). *VineCopula: Statistical Inference of Vine Copulas*, R package version 2.4.2.
- Park S and Baek C (2014). On multivariate GARCH model selection based on risk management, *Journal of the Korean Data and Information Science Society*, **25**, 1333–1343.
- Ryan JA and Ulrich JM (2020). *quantmod: Quantitative Financial Modelling Framework*, R package version 0.4-16.
- Sklar M (1959). Fonctions de repartition an dimensions et leurs marges, *Publications de l’Institut Statistique de l’Université de Paris*, **8**, 229–231.

Received July 26, 2021; Revised August 26, 2021; Accepted September 15, 2021

# 성근 바인 코플라 모형을 이용한 고차원 금융 자료의 VaR 추정

안광준<sup>a</sup>, 백창룡<sup>1,a</sup>

<sup>a</sup>성균관대학교 통계학과

---

## 요약

최대예상손실액(VaR)은 위험관리수단으로 금융에서 시장위험을 측정하는 대표적인 값이다. 본 논문에서는 다양한 자산으로 이루어진 고차원 금융자료에서 자산들 간의 의존성 구조를 잘 설명할 수 있는 성근 바인 코플라를 이용한 VaR 추정에 대해서 논의한다. 성근 바인 코플라는 정규 바인 코플라 모형에 벌점화를 적용한 방법으로 추정하는 모수의 개수를 벌점화를 통해 축소하는 방법이다. 모의 실험 결과 성근 바인 코플라를 이용한 VaR 추정이 더 작은 표본 외 예측오차를 줄일 수 있었다. 또한 최근 5년간의 코스피 60개 종목을 바탕으로 실시한 실증 자료 분석에서도 성근 바인 코플라 모형이 더 좋은 예측 성능을 보임을 확인할 수 있었다.

주요용어: 최대예상손실액, 성근 바인 코플라, 라소, 의존성 구조

---