Copula Processes in Modeling Economic Time Series and Comparison to GARCH Methods

Nathan Luksik
Duke University
nl146@duke.edu

Abstract

Copula processes are a recent extension of the increasingly popular usage of copulas in financial applications. Copulas allow learning marginal distributions separately from the multivariate dependence structure that joins them together into a density function, and copula processes extend this idea by allowing for the specification of time dependencies, via latent Gaussian processes. The purpose of this paper is to discuss and survey several implementations the recent and growing field of copula processes, particularly focusing on Gaussian copula processes and their applications in modeling and forecasting economic time series. The author then compares an example implementation of such copula processes to the popular GARCH model.

1 Introduction

Forecasting financial time series data is of great interest to economic modelers and the use of copulas in this field has been gaining popularity for the last few decades. More recently, the extension to copula processes has been explored and copula process-based methodologies have found success compared to current baseline and state-of-the-art forecasting methods. This paper will begin with a review of the necessary background on copulas and copula processes in Section 2, then will briefly explore a few interesting implementations of copula processes in Section 3, and will conclude in Section 4 with a comparison of Gaussian copula processes methods to a widely used baseline model known as GARCH.

2 Background

2.1 Copulas

Copulas can describe the dependency structure of any multivariate joint distribution $H(x_1, ..., x_n) = P(X_1 \leq x_1, ..., X_n \leq x_n)$ by taking each univariate random variable X_i and transforming it through its CDF F_i to get a uniform

random variable $U_i = F_i(X_i)$, then expressing the dependencies between these transformed values through the n-copula $C(u_1, ..., u_n)$.

Sklar's theorem [7] states that any joint distribution can be represented as a copula function of its univariate marginal distributions:

Sklar's Theorem: Let H be an n-dimensional distribution function with marginal distribution functions $F_1, F_2, ..., F_n$. Then there exists an n-copula C such that for all $(x_1, x_2, ..., x_n) \in [-\infty, \infty]^n$,

$$H(x_1, x_2, ..., x_n) = C(F_1(x_1), F_2(x_2), ..., F_n(x_n)) = C(u_1, u_2, ..., u_n). \quad (1)$$

If $F_1, F_2, ..., F_n$ are all continuous then C is unique. Otherwise, C is uniquely determined on Range $F_1 \times$ Range $F_2 \times \cdots \times$ Range F_n. Conversely, if C is an n-copula and $F_1, F_2, ..., F_n$ are distribution functions, then the function H is an n-dimensional distribution function with marginal distribution functions $F_1, F_2, ..., F_n$. Thus,

$$C(u_1, u_2, ..., u_n) = H(F_1^{(-1)}(u_1), F_2^{(-1)}(u_2), ..., F_n^{(-1)}(u_n)) \quad (2)$$

can be used to construct a copula.

Gaussian Copula: For example, one of the simplest and most widely used examples of a copula is the Gaussian copula, defined as

$$C(u_1, ..., u_n) = \Phi_\rho(\Phi^{-1}(u_1), ..., \Phi^{-1}(u_n)), \quad (3)$$

where Φ_ρ is a bivariate Gaussian CDF with correlation coefficient ρ, and Φ is the standard univariate Gaussian CDF [3].

Now, to specify time dependencies, we define copula processes below.

2.2 Copula Processes

Let W_t be a collection of random variables indexed by $t \in \mathcal{T}$, with marginal distribution functions F_t, and let $Q_t = F_t(W_t)$. Further, let μ be a stochastic process measure with marginal distribution functions G_t, and joint distribution function H. Then W_t is a copula process distributed with base measure μ, or $W_t \sim \mathcal{CP}(\mu)$, if and only if for all $n \in \mathbb{N}, a_i \in \mathbb{R}$,

$$P(\bigcap_{i=1}^{n} \{G_t^{(-1)}(Q_{t_i}) \leq a_i\}) = H_{t_1,t_2,...,t_n}(a_1, a_2, ..., a_n). \quad (4)$$

2.3 Stochastic Processes and Volatility

Stochastic Volatility refers to the idea that the volatility of asset prices varies and is not constant, which is assumed in the Black Scholes asset pricing model. To correct for this, stochastic volatility models are stochastic processes for which the variance itself is randomly distributed. The key feature of these models

is that volatility, being inherently unobservable and subject to independent random shocks, is not measurable with respect to observable information [1]. Processes for which the volatility is not constant are called heteroskedastic. How this leads into the ARCH and GARCH models discussed in this paper is explained in Section 4.1.

3 Survey

We now will introduce to the reader several kinds of copula processes along with some specific methodologies and applications. This is far from an exhaustive list, but it captures various interesting developments in the field and may serve as a starting point for the reader to explore the topic further.

3.1 Gaussian Copula Processes and Forecasting Volatility

In their 2010 paper, Ghahramani and Wilson [3] introduce copula processes and discuss Gaussian copula processes in particular:

W_t is Gaussian copula process (GCP) distributed if it is copula process distributed and the base measure μ is a Gaussian process. If there is a mapping Ψ such that

$$\Psi(W_t) \sim \mathcal{GP}(m(t), k(t, t')), \tag{5}$$

then we write

$$\Psi(W_t) \sim \mathcal{GCP}(\Psi, m(t), k(t, t')). \tag{6}$$

They then, to demonstrate its applications to machine learning and in finance, specifically regarding modeling and predicting volatility, they develop a practical demonstration of how a copula process can be used to generate samples of arbitrarily many random variables with desired marginals and dependencies. This stochastic volatility model they create is called the Gaussian Copula Process Volatility (GCPV) model. It learns the joint distributions of any number of random variables, as well as their marginals, and uses these to make predictions. They do so by fitting a GCP using a type of warped Gaussian Process.

Given a sequence of observations $y = (y_1, ..., y_n)^T$ at times $t = (t_1, ..., t_n)^T$, which are random variables with different latent standard deviations, one has n unobserved standard deviations $\sigma_1, ..., \sigma_n$ and the aim is to learn the correlation structure between these standard deviations and predict the the distribution of σ_* at some future time t_*.

The standard deviation function can be modeled as a GCP:

$$\sigma_t \sim \mathcal{GCP}(g^{-1}, 0, k(t, t')). \tag{7}$$

In particular,

$$f_t \sim \mathcal{GP}(m(t) = 0, k(t, t')) \tag{8}$$

$$\sigma(t) = g(f(t), \omega) \tag{9}$$

$$f(t) \sim \mathcal{N}(0, \sigma^2(t)) \tag{10}$$

where g is a monotonic warping function parameterized by ω. With GCPV, the form of g is learned so that $g^{-1}(\sigma_t)$ is best modeled by a Gaussian process. Learning g learns the marginal for each σ: $F(a) = \Psi(g^{-1}(a))$ for $a \in \mathbb{R}$. The authors assume the marginal distributions of σ_t are stationary over small time periods, and thus that the equations (7)-(9) hold true for each of these time periods.

The goal of this method is to infer $p(\sigma(t_*)|y, z)$ for $z = \{\theta, \omega\}$ and θ are the hyperparameters of the Gaussian process covariance function. This is done by sampling from

$$p(f_*|y, z) = \int p(f_*|f, \theta)p(f|y, z)df \tag{11}$$

and transforming these samples by g. To learn z, the authors find the $\hat{z}$ which maximizes the marginal likelihood,

$$p(y|z) = \int p(y|f, \omega)p(f|\theta)df. \tag{12}$$

For many functions g, (11) and (12) are intractible, so the authors use the Laplace approximation and also used Markov Chain Monte Carlo to make predictions of σ_* by sampling from $p(f_*|y, z)$ and to determine z.

In their experiments, the authors predict the latent standard deviations σ of observations y at times t, and also σ_* at unobserved times t_* to forecast volatility. They used simulated data as well as real historical financial data of currency exchange rates. How their forecasting compares to benchmarks like GARCH is discussed in Section 4.2.

3.2 Gaussian Copula Processes with RNN for Financial Time Series

In their 2019 paper, Salinas et al [6] combine a recurrent neural network (RNN) time series model with a GCP output model with a low-rank covariance structure to reduce computational complexity and to handle non-Gaussian marginal distributions for multivariate forecasting. By reducing the number of parameters, the authors were able to model time-varying correlations of thousands of time series, and with a higher accuracy than the current state-of-the-art deep learning methods. The two main issues preventing other high-dimensional models from being estimated well are the the $O(N^2)$ scaling of the number of parameters required to express the covariance matrix, as well as the fact that the magnitudes of time series can vary drastically between different series of the same data set.

To alleviate the first issue, his paper utilizes the low-rank-plus-diagonal structure of the Factor Analysis model [5] in combination with a GCP to an RNN to jointly learn the temporal dynamics and time-varying covariance structure while reducing the number of parameters that need to be estimated. To deal with the second issue, this paper models each time series' marginal distributions separately with a nonparametric estimate of the CDF, and use the

CDF as the marginal transformation of a Gaussian copula to decouple the estimation of marginal distributions from temporal dynamics and the dependency structure.

The authors experiment on synthetic data as well as on several real world data sets, and find that their method provides significant forecasting accuracy improvements over their baselines of a multivariate linear vector auto-regressive model (VAR), GARCH (which is defined and explained in section 4.1), several RNN architectures, and some state-of-the-art deep learning models (DeepAR and MQCNN). They computed 400 samples for each model and evaluated the multi-step accuracy using the continuous ranked probability score metric which measures the accuracy of the predicted distribution. Exactly how this model compares to GARCH methods is described in Section 4.2.

3.3 Gaussian Processes with Vine Copulas

In their 2013 paper, Lopez-Paz and Ghahramani [4] discuss the usage of vine copulas, which specify a factorization of any high-dimensional copula density into a product of conditional bivariate copulas. Unfortunately, some of the conditional dependencies in these copulas are typically ignored when constructing the vine, which yields estimates too simplistic to accurately model real world data. The authors of this paper present a method for the estimation of fully conditional vines using Gaussian processes which they call GPVINE. This method is based around discovering latent functions that specify the shape of conditional copula given its conditioning variables.

Specifically, vine copulas are hierarchical graphical models that factorize a d-dimensional copula density $c(u_1, ..., u_d)$ into a product of $d(d-1)/2$ bivariate conditional copula densities. A vine $\mathcal{V}$ is constructed by forming a nested set of $d-1$ undirected trees, in which each of their edges corresponds to a conditional bivariate copula density. Vine distributions require the calculating of marginal conditional CDFs and conditional bivariate copula densities, and the number of variables to condition on increases as one progresses deeper in the vine hierarchy.

The authors' approach to estimating conditional bivariate copulas yields equations which cannot be computed analytically, so they approximate them using Expectation Propagation (EP), but this has a cost of $O(N^3)$. Therefore, the authors use the FITC approximation for Gaussian Processes [8], which allow the EP method to run at a cost of $O(NN_0^2)$ ($N_0 << N$ being the number of training points or pseudo-inputs).

The authors experimented with GPVINE against SVINE (a vine model based on the simplifying assumption described above where any conditional dependencies in the bivariate copulas are just ignored) and another model called MLLVINE (which handles conditional dependencies with respect to only one single scalar variable). Experiments with both synthetic and real-world data show that GPVINE frequently has better predictive performance than both the baseline method SVINE that ignores conditional dependencies and the current state-of-the-art alternative MLLVINE.

4 GCP vs GARCH

We have thus far discussed copula processes, particularly Gaussian copula processes, and several novel methods which utilize them to perform forecasting and modeling of time series data with greater performance and accuracy than baseline as well as some state-of-the-art methods. In this section, we will explain more about some of these other methods and compare them to the GCP-based methods described above.

4.1 GARCH

Understanding volatility in financial asset returns is crucial for hedging, risk management, and portfolio optimization, and invaluable to this pursuit is explicitly modeling time-varying second-order moments [2]. Important in characterizing such variances was the development of the Autoregressive Conditional Heteroskedasticity (ARCH) model.

Heteroskedasticity refers to the volatility of a process not being constant. Volatility can be interpreted as the square root of the conditional variance of the log return process given by its previous values [9]. Hence, if P_t is a time series value at time t, the log return is given by $X_t = \log P_{t+1} - \log P_t$ and the volatility is σ_t, where $\sigma_t^2 = Var[X_t^2|\mathcal{F}_{t-1}]$ (for $\mathcal{F}_{t-1}$ the σ-algebra generated by $X_0, X_1, ..., X_{t-1}$).

The ARCH model described forecast variance in terms of current observations by taking the averages of past squared forecast errors. The log returns are modeled as white noise multiplied by volatility:

$$X_t = \epsilon_t \sigma_t \qquad \sigma_t^2 = \omega + \alpha_1 X_{t-1}^2 + ... + \alpha_p X_{t-p}^2 \tag{13}$$

Note that ϵ_t are iid with expectation 0 and variance 1, and are assumed independent from σ_k for all $k \leq t$. Further, we have lag length $p \geq 0$, and σ_t^2 is ensured to be positive since $\omega, \alpha_i \geq 0$ for all i.

Making generalizations to different weighting schemes led to the inception of the widely used Generalized ARCH (GARCH) model.

GARCH generalizes the autoregressive ARCH to an autoregressive moving average model whose forecast variance is a weighted average of three different variance forecasts: a constant variance corresponding to the long run average, the forecast that was made in the previous period, and new information that was not available when the previous forecast was made. Hence, the variance forecast is based off of one period of information. For the general case GARCH(p,q):

$$X_t = \epsilon_t \sigma_t \qquad \sigma_t^2 = \omega + \alpha_1 X_{t-1}^2 + ... + \alpha_p X_{t-p}^2 + \beta_1 \sigma_{t-p}^2 + ... + \beta_q \sigma_{t-q}^2. \tag{14}$$

The simpler model GARCH(1,1) is generally favored by economists due to their simpler implementation; they are given by stochastic difference equations in discrete time, and the likelihood function is then easier to handle than continuous-time models [9]. For GARCH(1,1),

$$X_t = \epsilon_t \sigma_t \qquad \sigma_t^2 = \omega + \alpha_1 X_{t-1}^2 + \beta_1 \sigma_{t-p}^2. \tag{15}$$

	exchange	solar	electricity	traffic
GARCH	0.024	0.928	0.291	0.426
GCP	0.008	0.371	0.056	0.133

Table 1: GARCH vs GCP method

GARCH has been further expanded and generalized into many models over the years since its inception, and the GARCH family has been a widely used benchmark to model and forecast volatility to this day.

4.2 GCP and GARCH Comparison

The GCP methods discussed in Section 3 both were experimented against other, similar state-of-the-art methods, but were also compared to GARCH as a baseline.

In Ghahramani and Wilson's 2010 paper on copula processes [3], their GCPV model outperformed GARCH with the simulated data, especially in regions of low volatility. GCPV more accurately captures the dependencies between σ at different times than GARCH. For the experiments with real financial data (Deutchmark vs the Great Britain Pound from 1984-1992), GARCH overestimated volatility, while GCPV was consistently a little closer to the real values.

Then, in Salinas et al [6], their GCP plus RNN model was tested against GARCH (among other models and baselines) in multiple real-world data sets: exchange data, solar data, electricity data, and traffic data. The models were scored for each data set based on CRPS (Continuous Ranked Probability Score) accuracy metrics. Note that scores under this metric are minimized when the predictive distribution is equal to the real distribution from which the data is drawn. That is, a smaller score is better. As we can see in Table 1 above, the GCP consistently scores significantly lower (that is, more accurate) than the baseline GARCH method.

5 Conclusion

This paper discussed copula processes, focusing on Gaussian copula processes, and explored several methods for modeling and forecasting financial time series. It then compared these methods to the widely used GARCH baseline, and demonstrated that, for the implementations described above and the data that was experimented upon, Gaussian copula processes-based methods appear to be more accurate than GARCH. This brief survey serves as a starting point for the reader interested in further exploring various implementations of copula processes in financial time series forecasting.

References

[1] Andersen, T., and Benzoni, L. Stochastic volatility. Federal Reserve Bank of Chicago (2009).

[2] Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3): 307-327, 1986.

[3] Ghahramani, Z., and Wilson, A. G. Copula processes. Cambridge Press (2010).

[4] Lopez-Paz, D., and Ghahramani, J. Gaussian process vine copulas for multivariate dependence. Proceedings of the 30th International Conference on Ma- chine Learning, Atlanta, Georgia, USA, 2013.

[5] Rubin, D., and Thayer, D. Em algorithms for ml factor analysis. Psychometrika 47, pages 69–76 (1982).

[6] Salinas, D. e. a. High-dimensional multivariate forecasting with low-rank gaussian copula processes. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

[7] Sklar, M. Fonctions de répartition À n dimensions et leurs marges. Universite Paris (1959).

[8] Snelson, E., and Ghahramani, Z. Sparse gaussian processes using pseudo-inputs. Advances in Neural Information Processing Systems (2006).

[9] Williams, B. Garch(1,1) models. University of California at Berkeley, 2011.