

ABSTRACT

Title of Dissertation: **MULTIVARIATE BILATERAL GAMMA
PROCESS WITH FINANCIAL APPLICATION
AND MACHINE LEARNING IN CORPORATE
BOND MARKET**

Yiran Zhang
Doctor of Philosophy, 2022

Dissertation Directed by: **Professor Dilip B. Madan**
Department of Finance

This dissertation consists of three essays. Chapter 1 is titled “Calibration for multivariate bilateral gamma model”. In this chapter, we discuss the multivariate bilateral gamma (MBG) model, which is an extension of the multivariate variance gamma (MVG) model that consists of predetermined Bilateral Gamma (BG) marginals. This model is used to model financial asset returns that have excess kurtosis, negative skewness, and asymmetry in the upward and downward motions. An efficient Monte Carlo simulation schema of this process is devised. Moreover, we propose an estimation procedure for the MBG model based on the Continuum Generalized Method of Moments (CGMM) and reparameterization of its correlation matrix. We compare this model with the full Gaussian copula (FGC) model by fitting it to the US equity data in the Dow Jones index, and the result indicates that the MBG model outperforms significantly. Chapter 2 is titled “Pairs trading strategies: Distance, cointegration, copula, and MBG method”. In this chapter, we propose a new method of building pairs trading strategy that uses MBG to model the de-

pendency structure of stock pairs. The trading signal is then built based on the cumulative mispricing indexes, which are calculated using the conditional probability of the MBG distribution. We also conduct a comprehensive study on the performance of four different pairs trading strategies—the distance, cointegration, copula, and MBG methods—on the US equity market from 2016 to 2019. The result shows that the MBG method has the highest excess return and records the best risk-adjusted performance. Chapter 3 is titled “Predictability of corporate bond returns with machine learning.” This chapter performs a comparative analysis on machine learning methods for the predictability of corporate bonds returns. We found that machine learning models, especially ensemble learning methods like boosting, substantially improve the out-of-sample performance of stock and bond characteristics in predicting future bond returns. We also show that portfolios constructed using the prediction of machine learning models contribute to economic gains. To investigate which features the models rely on when making a prediction, we discuss three methods for calculating feature importance, including SHapley Additive exPlanations (SHAP) based on Shapley value. This method is model agnostic, consistent, and can provide an explanation for each observation. We also discuss the performance of the parsimonious model with the 15 most important features selected by SHAP.

MULTIVARIATE BILATERAL GAMMA PROCESS WITH
FINANCIAL APPLICATION AND MACHINE LEARNING IN
CORPORATE BOND MARKET

by

Yiran Zhang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Advisory Committee:
Professor Dilip B. Madan, Chair/Advisor
Professor Paul J. Smith
Professor Benjamin Kadem
Professor Leonid Koralov
Professor Mark Loewenstein

© Copyright by
Yiran Zhang
2022

Acknowledgments

First and foremost, I am extremely grateful to my supervisor, Dilip B. Madan for his invaluable advice, continuous support, and patience during my Ph.D. study. His immense knowledge and plentiful experience have encouraged me throughout my academic research and daily life. Without these, this dissertation would not have been possible.

I would like to express my deepest appreciation to Professor Paul Smith, Professor Benjamin Kedem, Professor Leonid Koralov, and Professor Mark Loewenstein for accepting to serve on my thesis committee and sparing their invaluable time reviewing my dissertation. Many Thanks for their encourage and support for all aspects in my graduate study.

I would like to offer special thanks to Professor Peter Carr, who helped and encouraged me and my husband a lot during our study at New York University and through our career path in quantitative finance. Although no longer with us, he continues to inspire us by his example and dedication.

I would like to extend my sincere gratitude to the group members in Mathematics Finance RIT for our discussion. Special thanks to Sahil Chopra, Yoshihiro Shirai, Shahnawaz Khalid, Zhang Zhang, and Ran Li.

I would like to express my sincere gratitude to my friends—Xuze Zhang, Yifan Yang, Nathan Yu, Xiaoyu Zhou, Yi Zhou, Yunjiang Ge, Xiaocui Zhang—for the wonder-

ful graduate life at College Park.

I owe my deepest thanks to my family—my mother, father, and my grandparents—who raised me and have always stood by and guided me through my life. Words cannot express the gratitude I owe them.

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever. I am sure this enumeration is far from being complete for the people that I own a debt of gratitude to; any inadvertent omission on the list is my fault.

Finally, I would like to thank my son, George Wang for bringing joys to my life, and my husband, Daoxing Wang, for his unwavering support.

Table of Contents

Acknowledgments	ii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
Chapter 1: Calibration of Multivariate Bilateral Gamma Model	1
1.1 Introduction	1
1.2 Probability Background	5
1.2.1 Lévy processes	5
1.2.2 Variance gamma process	8
1.2.3 Bilateral gamma process	12
1.2.4 Full Gaussian copula model	16
1.3 Multivariate Bilateral Gamma Model and Inference	21
1.3.1 Properties of the MBG model	21
1.3.2 Simulation for the MBG model	25
1.3.3 Estimation procedure for the MBG model	26
1.4 Performance on Fitting Stock Returns	39
1.5 Conclusion and Future Research	42
1.6 Appendix	44
1.6.1 The Proof of Lemma 1.3.1	44
1.6.2 The Proof of Theorem 1.3.1	47
Chapter 2: Pairs Trading Strategies: Distance, Cointegration, Copula, and Multivariate Bilateral Gamma Method	51
2.1 Introduction	51
2.2 Literature Review	54
2.3 Research Method	58
2.3.1 Distance method	59
2.3.2 Cointegration method	61
2.3.3 Copula method	65
2.3.4 MBG method	71
2.4 Data and Empirical Result	74
2.4.1 Data	74

2.4.2	Transaction cost	74
2.4.3	Return calculation methodology	75
2.4.4	Descriptive statistics of portfolio excess return	76
2.4.5	Risk-adjusted performance	78
2.4.6	Properties of each pair trade	82
2.4.7	Risk profile	86
2.4.8	Sensitivity analysis to hyperparameters	89
2.5	Conclusion and future research	93
2.6	Appendix	95
Chapter 3:	Predictability of Corporate Bond Returns with Machine Learning	96
3.1	Introduction	96
3.2	Research Method	102
3.2.1	Methodology	102
3.2.2	Training, validation, and test sets	109
3.2.3	Performance evaluation metric	111
3.2.4	SHapley Additive exPlanations (SHAP)	112
3.3	Data and Variable Definitions	116
3.3.1	Corporate bond data	116
3.3.2	Corporate bond return	119
3.3.3	Bond characteristics	121
3.3.4	Stock characteristics	130
3.3.5	Joining stock and bond characteristics	130
3.3.6	Dealing with missing data	131
3.3.7	Exploratory data analysis of our dataset	133
3.3.8	Bond filter rules	135
3.4	Predictability of bond returns	136
3.4.1	Out-of-sample predictability of bond returns using bond characteristics	136
3.4.2	Out-of-sample predictability of bond returns using bond and equity characteristics	139
3.5	Feature Importance	142
3.5.1	The naive approach	143
3.5.2	Feature importance using reduction in R-squared value	145
3.5.3	Feature importance using SHAP	146
3.5.4	Feature importance for bond and equity characteristics using SHAP	152
3.6	Machine Learning Portfolio	155
3.7	Parsimonious Model	162
3.8	Conclusion and future research	165

List of Tables

1.1	CGMM 3-dimension estimation result.	38
1.2	CGMM 4-dimension estimation result.	39
1.3	MBG and FGC comparison using Anderson-Darling distance	42
2.1	Mean return	77
2.2	Risk-adjusted return	82
2.3	Return of each trade	84
2.4	Trade convergence analysis	86
2.5	Fama-French	88
2.6	Shapiro–Wilk test for portfolio excess return	95
3.1	Credit ratings.	122
3.2	Missing data analysis	132
3.3	Summary statistics for each variable	135
3.5	Optimal hyperparameters for bond features model	138
3.4	Out-of-sample R^2 for bonds features	138
3.6	Hyperparameters search space for bond features only model	139
3.8	Hyperparameter search space for bond and equity features model	141
3.7	Out-of-sample R^2 for bond and equity features	141
3.9	Optimal hyperparameters for bond and equity features model	142
3.10	Key performance metric	157
3.11	Average precision of our recommendation	157
3.12	Performances of machine learning portfolios	159
3.13	Out-of-sample R^2 for parsimonious model	163
3.14	Optimal hyperparameters for parsimonious model	164
3.15	Hyperparameters search space for parsimonious model	165

List of Figures

2.1	Cointegrated and non-conintegrated stock pairs	65
2.2	Cumulative return	79
2.3	Return distribution for each trade (with transaction cost).	84
2.4	Return distribution for each converged trade (with transaction cost)	85
2.5	Exit status for different strategies	86
2.6	Average number of trades per selected pair	90
2.7	Traded pairs / Selected pairs	91
2.8	Percent of convergence and days to converge	91
2.9	Sensitivity analysis of MBG method	92
3.1	Regression tree example	105
3.2	Walk forward cross validation	111
3.3	Density plot for time-to-maturity when bonds issue	133
3.4	Rating of bonds	134
3.5	Feature importance of models from scikit-learn default	144
3.6	Feature importance of models based on reduction of R^2	146
3.7	Feature importance based on SHAP(bond only)	148
3.8	Feature importance for single observation(XGBoost)	150
3.9	Overall SHAP feature importance(bond only)	151
3.10	Feature importance of models based on SHAP(bond and equity)	154
3.11	Overall SHAP Feature Importance for Bond and Equity	155
3.12	Cumulative return for XGBoost	156
3.13	Cumulative return for ML long only strategy	160
3.14	Cumulative return for ML long short strategy	161
3.15	Cumulative return for ML short only strategy	161

Chapter 1: Calibration of Multivariate Bilateral Gamma Model

1.1 Introduction

It is well-known that the normal distribution is very important in probability theory and statistical applications. Since [Black and Scholes \(1973\)](#) and [Merton \(1973\)](#) assumed the normal distribution of stock returns and used the Brownian motion to model the data on financial return from the stock and option markets, there has been considerable interest in improving upon their results. The subsequent development can be roughly classified into two primary directions: considering other distributions than normal distributions and extending from the univariate distributions to those with a high dimensional space.

After observing the excess kurtosis in the distribution of financial returns, [Madan and Seneta \(1990\)](#) proposed a symmetric variance gamma model by considering the variance in the models of [Black and Scholes \(1973\)](#) and [Merton \(1973\)](#) as gamma-distributed. Later [Madan et al. \(1998\)](#) extended from the symmetric variance gamma model to the asymmetric variance gamma model. The new model can be used to model the volatility skewness that was often observed in the risk-neutral distributions implied by option prices. More recently, [Madan and Wang \(2017\)](#), and [Madan et al. \(2017\)](#) recommended the use of the bilateral gamma (BG) model introduced by [Küchler and Tappe \(2008\)](#) after observing the asymmetry in the upward and downward motions of the stock: upward

motions have more frequent and smaller moves in comparison than downward motions.

While there has been extensive research on using the univariate distributions to model financial returns, research on using the multivariate distributions to model such data is still less explored. With the increase in the popularity of exchange-traded funds and multi-asset derivatives, developing methods to model dependent stock prices is necessary. [Dhaene et al. \(2014\)](#) extended the univariate Black–Scholes model to the multivariate case. Their extension uses a multivariate normal dependency structure to model the joint dynamics of stock returns and utilizes a log-normal distribution for stock prices. However, the main drawback to this approach is that a multidimensional Brownian motion is too simple to capture the sophisticated dynamics of assets in financial markets. Based on observed data, asset returns often show negative skewness and excess kurtosis and do not follow a normal distribution. A more realistic approach is the full Gaussian copula (FGC) model ([Malevergne and Sornette, 2003](#)), under which the joint distribution is generated from several marginal distributions where the copula function is implied by a multivariate Gaussian distribution. Since this model allows the separation of dependence structure from the marginal distributions, one can select a sophisticated marginal distribution such as BG marginal, thereby allowing for skewness and excess kurtosis. The FGC approach, however, cannot effectively model tail dependence ([Renard and Lang, 2007](#)). Empirical studies indicate that stock returns for similar companies often exhibit strong tail dependence for significant negative movements. Unfortunately, the FGC model cannot capture this effect, and this results in underestimating the probability of extreme losses. This inability to model tail dependence shares part of the blame for the 2008 global financial crisis. To overcome this deficiency, models such as multivariate variance gamma

(MVG) and multivariate bilateral gamma (MBG), have been proposed. MVG model, presented in [Buchmann et al. \(2019\)](#), is a multivariate Lévy process that has a Lévy density with full support and is consistent with predetermined Variance Gamma (VG) marginals that have distinct parameters for controlling kurtosis levels for each dimension. Similar to the univariate case, the asymmetry in the upward and downward motions of a financial asset has been observed in multivariate cases. [Madan \(2020\)](#) extended the construction of MVG to consist of predetermined BG marginals to take advantage of the BG distribution in modeling financial asset returns. The resulting model, called MBG, is also a Lévy process that has a Lévy density with full support and is consistent with predetermined BG marginals that have different parameters for controlling kurtosis for each dimension.

Estimation is the process of finding model parameters that provide the best fit to the observed data, and there are already some widely used methods. Examples of such methods are maximum likelihood estimation (MLE), method of moment (MOM), and generalized method of moment (GMM). For a model with a known joint density function, the MLE is generally used in the estimation procedure because its estimators are known to be asymptotically unbiased and efficient. However, like most Lévy processes, MBG does not have a close form density function. Under this scenario, MLE method works theoretically but leads to complicated numerical computation in practice. In this case, inference procedures based on characteristic functions (CF) rather than densities are encouraged. Different GMM-like estimators based on CF have been employed to solve such problems, and the simplest of these consist of applying the GMM procedure to a discrete subset of moment conditions ([Feuerverger and McDunnough, 1981](#)). The advantage of this method is that the asymptotic variance of the estimator can be made arbitrarily

close to the Cramer–Rao bound by growing the grid finer. However, the covariance matrix of the moment condition becomes singular when the number of points in the grid exceeds that of observations. To address this limitation, [Carrasco and Florens \(2000\)](#) extended the GMM to construct the continuum-GMM (CGMM) by using the continuum of moment conditions instead of selecting finite grid points. Because of the infinity of moments, the covariance operator is nearly singular, and a penalty term α is introduced to stabilize the inverse.

In the financial market application—especially when modeling multivariate assets returns—each observation is usually a historical return; hence, we have limited data points. Additionally, since the market condition is constantly evolving, only the most recent historical information should be used to accurately capture the current. To do this, we need to exclude the data from the distant past, which results in even less data. Therefore, we adopt the CGMM method when calibrating the MBG model to achieve a more efficient estimator. However, supposing the CGMM procedure is applied directly, we cannot guarantee that the correlation matrix in the MBG model is positive semidefinite in each iteration. To address this issue, in the proposed estimation procedure, we reparametrize the correlation matrix as a vector that is unrestricted in $\mathbb{R}^N$, following [Archakov and Hansen \(2021\)](#), and apply the CGMM on the vector instead. Additionally, we observe that two vastly different reparametrized correlation matrices could result in similar correlation matrices in practice. This behavior would make the optimization routine unstable, especially in the higher dimensional case (larger than four). To solve this problem, we further introduce a penalty term that adds to the CGMM objective function to control the norm of the reparametrized correlation matrix. This penalty term helps to

stabilize the optimization routine. Digital moment estimation (DME), first introduced by [Madan \(2015\)](#), was later used by [Madan and Schoutens \(2022\)](#) to estimate the MBG process. Despite its ease of implementation when compared with CGMM, CGMM provides a more accurate estimation. In this chapter, we also compare the performance of fitting the MBG and FGC using stock returns in the Dow Jones Index and conclude that MBG significantly outperforms the FGC.

This chapter is organized as follows. In Section 2, we give a brief overview of Lévy processes and introduce univariate Lévy processes, such as the VG and BG. We then introduce the FGC, a widely used model in the financial industry to model multivariate assets returns. We also introduce the MBG model, which benefits from the univariate BG process, allowing for skewness and excess kurtosis. The MBG model also nicely captures the tail dependency of financial assets returns by introducing non Gaussian dependence structures. The Monte Carlo simulation schema and our proposed calibration procedure for the MBG model are discussed in Section 3. In the same section, we present the results of fitting the MBG to US equity data in the Dow Jones index. Section 4 presents the conclusion.

1.2 Probability Background

1.2.1 Lévy processes

The Lévy process is named after the mathematician Paul Lévy, the theory pioneer. One of the simplest and the most famous Lévy processes is the Brownian motion, a stochastic process with independent, stationary, and normally distributed increments.

Since the 1900s, after Louis Bachelier used this process to model stock options in his Ph.D. thesis, people have developed financial models driven by Brownian motion. An excellent example of such a model is the Black–Scholes model, developed in 1973 by Fischer Black, Robert Merton, and Myron Scholes. Despite its popularity, one of the main problems with the Black–Scholes is that actual financial data suggested that the log return of assets does not follow the Normal distribution and is usually negatively skewed and leptokurtic. To overcome this limitation, researchers in the 1980s and 1990s proposed Lévy processes that are based on more general distribution. Examples of such models that take into account the skewness and excess kurtosis are the VG, the normal inverse Gaussian, and the CGMY (named after Carr, Geman, Madan, and Yor). Loosely speaking, Lévy processes are stochastic processes with independent and stationary increments and can be viewed as the continuous-time analog of a random walk. The formal definition is given in the following:

Definition 1.2.1 (Lévy process). A càdlàg stochastic process $(X_t)_{t \geq 0}$ on $(\Omega, \mathcal{F}, \mathbb{P})$ with values in $\mathbb{R}^d$ such that $X_0 = 0$ is called a Lévy process if it possesses the following properties:

- Independent increments: for every increasing sequence of times $t_0 \dots t_n$, the random variables $X_{t_0}, X_{t_1} - X_{t_0}, \dots, X_{t_n} - X_{t_{n-1}}$ are independent.
- Stationary increments: the law of $X_{t+h} - X_t$ does not depend on t .
- Stochastic continuity: $\forall \varepsilon > 0, \lim_{h \rightarrow 0} \mathbb{P}(|X_{t+h} - X_t| \geq \varepsilon) = 0$.

Definition 1.2.2 (Infinitely divisible). Let $\phi(u)$ be the characteristic function of a distribution F . We can say that the F is infinitely divisible, if for all integer $n > 0$, $\phi(u)$ can

be represented as the n -th power of a characteristic function.

The following proposition provides the link between the Lévy process and infinitely divisible distribution.

Proposition 1.2.1 (Infinite divisibility and Lévy processes). If X_t is a Lévy process, then for every $t \geq 0$, X_t has an infinitely divisible distribution. Conversely, if F is an infinitely divisible distribution, there exists a Lévy process X_t such that the distribution of X_1 is F .

Let $\Phi_{X_t}(z)$ be the characteristic function of a Lévy process X_t . If we define $\psi(u) = \log(\Phi(u))$, often called characteristic exponent, then we have the following fact:

$$\Phi_{X_t}(u) = E(e^{iuX_t}) = e^{t\psi_{X_1}(u)}$$

where $\psi_{X_1}(u)$ is the characteristic exponent of the Lévy process at a single time $t =$

1. The characteristic exponent leads to the famous Lévy–Khintchine formula, using the characteristic exponent to describe all Lévy processes.

Definition 1.2.3 (Lévy–Khintchine formula). Let X_t be a Lévy process and $\psi(u)$ be its characteristic exponent; Lévy–Khintchine formula express the characteristic exponent $\psi(u)$ as follows:

$$\psi(u) = i\gamma u - \frac{1}{2}\sigma^2 u^2 + \int_{-\infty}^{+\infty} (\exp(iux) - 1 - iux1_{\{|x|<1\}}) \nu(dx)$$

where $\gamma \in \mathbb{R}$, $\sigma^2 \geq 0$ and ν is a measure on $\mathbb{R} \setminus \{0\}$ with $\int_{-\infty}^{+\infty} (1 \wedge x^2) \nu(dx) < \infty$.

From the Lévy–Khintchine formula, we can see that a Lévy process consists of three parts: a linear drift with rate γ , a Brownian part with volatility σ , and a Lévy jump process

with the Lévy measure $\nu(dx)$. Hence, each Lévy process can be uniquely determined by the Lévy–Khintchine triplet $(\gamma, \sigma^2, \nu(dx))$. For example, a Lévy process with Lévy triplet equal to $(0, \sigma^2, 0)$ is a standard Brownian motion. If $(\gamma, 0, \nu(dx))$, the process has no diffusion term, then the Lévy process is called a pure jump process. For a process with Lévy triplet $(0, 0, \lambda\delta(1))$, when $\delta(1)$ is the Dirac function at 1, then the process becomes a Poisson process. The gamma process that is written as $\Gamma(t; \gamma, \lambda)$ is also considered a pure jump Lévy process with independent gamma-distributed increment. The Lévy measure of gamma process is $\nu(x) = \gamma x^{-1} \exp(-\lambda x)$, for $x > 0$. Since the VG process is constructed as the difference of two independent gamma processes, and the Lévy process is closed under addition, the VG process introduced by [Madan and Seneta \(1990\)](#) is also a Lévy process. In the following sections, we will deal with the processes constructed by the difference of two independent gamma processes.

1.2.2 Variance gamma process

VG distribution gets its name and formulation from the normal variance-mean mixture, where the mixing density is the gamma distribution. VG distribution is generally used to model asset returns in the financial market, as the data display skewness and kurtosis different from the normal distribution. The class of VG distribution was introduced initially in the 1980s by [Madan and Seneta \(1987\)](#). [Madan and Seneta \(1990\)](#) proposed a symmetric VG in 1990, a model with two parameters that control the variance and kurtosis of the distribution. Later, [Madan et al. \(1998\)](#) introduced a generalized three-parameter VG that has an additional parameter for controlling the skewness. In the article below,

VG refers to the generalized VG introduced by [Madan et al. \(1998\)](#).

VG distribution can be constructed by mixing the Normal distribution with Gamma random variable. Let $G \sim \text{Gamma}(\frac{1}{\nu}, \frac{1}{\nu})$, then

$$X \sim \text{Normal}(\theta G, \sigma^2 G) \text{ follows a VG distribution denoted as } \text{VG}(\sigma, \nu, \theta).$$

The three parameters include a real number θ , which is the mean of the Normal distribution; a positive number, σ^2 , which represents the variance of the Normal distribution; and a positive number, ν , from the Gamma random variable G .

Definition 1.2.4 (Variance Gamma Distribution). The characteristic function of $\text{VG}(\sigma, \nu, \theta)$ is:

$$\phi_{\text{VG}}(u; \sigma, \nu, \theta) = \left(1 - iu\theta\nu + \frac{1}{2}\sigma^2\nu u^2\right)^{-1/\nu}.$$

From Definition 1.2.4, we know that the VG distribution is infinitely divisible. Based on the definition of Lévy process, we can construct the VG process as follows:

Definition 1.2.5 (Variance Gamma Process). The VG process $X^{(\text{VG})} = \{X_t^{(\text{VG})}, t \geq 0\}$ is a process which starts at zero, has independent and stationary increments and for which the increment $X_{s+t}^{(\text{VG})} - X_s^{(\text{VG})}$ follows a $\text{VG}(\sigma\sqrt{t}, \nu/t, t\theta)$ law over the time interval $[s, t + s]$. Clearly (take $t = 1, s = 0$ and note that $X_0^{(\text{VG})} = 0$), we have

$$\begin{aligned} E \left[\exp \left(iu X_t^{(\text{VG})} \right) \right] &= (\phi_{\text{VG}}(u; \sigma, \nu, \theta))^t \\ &= \left(1 - iu\theta\nu + \frac{1}{2}\sigma^2\nu u^2\right)^{-t/\nu}. \end{aligned}$$

Alternatively, the VG process can be constructed as the difference between two independent Gamma processes, detailed by [Madan et al. \(1998\)](#). Distinguished from

other Lévy processes, the VG process has finite variation; therefore, we can express the VG process as the difference of two increasing processes—the Gamma processes. In the context of financial return data, the first describes the price increases, whereas the second describes the price decreases.

Let Z be a random variable following VG distribution $Z \sim \text{VG}(c, b_p, b_n)$. As shown below, Z can be constructed as the difference of two independent Gamma random variables X and Y . If we have X and Y as gamma distributed with

$$X \sim \Gamma(c, b_p), \quad Y \sim \Gamma(c, b_n),$$

where c is considered as the shape parameter in Gamma distribution, and b_n, b_p as the scale parameter in Gamma distribution. Then we have Z

$$Z = X - Y \sim \text{VG}(c, b_p, b_n).$$

We use the convolution property to sum the two independent random variables X and $-Y$ leads to

$$\phi_Z(u) = (1 - iub_p)^{-c} (1 + iub_n)^{-c} = (1 - iu(b_p - b_n) + u^2 b_p b_n)^{-c}.$$

From Lévy–Khintchine formula, for X and $-Y$, the Lévy measure of VG in Lévy triplet is given by

$$v(dx) = \left(\frac{c}{x} e^{-\frac{x}{b_p}} \mathbb{I}_{(0, \infty)}(x) + \frac{c}{|x|} e^{-\frac{|x|}{b_n}} \mathbb{I}_{(-\infty, 0)}(x) \right) dx.$$

The VG density function is given by

$$f_{\text{VG}}(x; c, b_p, b_n)(x) = \frac{1}{\sqrt{\pi}\Gamma(c)(b_p b_n)^c} \exp\left(\frac{(b_n - b_p)x}{2b_p b_n}\right) \left(\frac{b_p b_n |x|}{b_p + b_n}\right)^{c-\frac{1}{2}} K_{c-\frac{1}{2}}\left(\frac{b_p b_n |x|}{b_p + b_n}\right),$$

where $K_v(x)$ denotes the modified Bessel function of the third kind with index v , and $\Gamma(x)$ denotes the Gamma function.

We can see that this density function is highly complex and has no close form. It will be challenging and time-consuming to use the MLE to directly estimate the parameters. Therefore, we use fast Fourier transform(FFT) to convert the characteristic function to densities and then use the MLE to estimate the parameters.

The two parameterizations, $\text{VG}(\sigma, \nu, \theta)$ and $\text{VG}(c, b_p, b_n)$, could be connected with the following formulas:

$$\nu = \frac{1}{c}, \quad \sigma^2 = 2cb_p b_n, \quad \theta = c(b_p - b_n).$$

From Definition 1.2.4, the univariate characteristic function of VG distribution, we can quickly obtain a multivariate extension of the VG distribution.

Definition 1.2.6. Let X denote a vector of random variables distributed conditional on the non-negative Gamma random variable G as a multivariate normal with mean vector θG and variance-covariance matrix ΣG . Note that G has unit mean and variance ν . By conditioning on G , the joint characteristic function of X , is easily defined as:

$$\phi_X(u) = \left(\frac{1}{1 - iu^\top \theta \nu + \frac{\nu}{2} u^\top \Sigma u} \right)^{\frac{1}{\nu}},$$

where $u^\top$ denotes the transpose of u .

In summary, VG gained popularity in handling the skewness and excess kurtosis displayed from the financial data. In VG distribution, the parameter σ accounts for its variance, similar to its role in Gaussian distribution. The parameter, θ , controls the skewness of the VG distribution; the distribution is skewed right with positive θ and skewed left with negative θ , with $\theta = 0$, the VG is symmetric. The parameter, ν , controls the kurtosis of the distribution; the kurtosis of the distribution increase as ν decreases.

1.2.3 Bilateral gamma process

The BG model, introduced by [Küchler and Tappe \(2008\)](#)), contains four parameters and can be considered a generalization of the VG model of [Madan and Seneta \(1990\)](#) and [Madan et al. \(1998\)](#). The VG model, originally presented as a log-normal density with a mean and variance that follow a Gamma distribution, can also be expressed as the difference of two Gamma processes with the same speed parameters but different scales. After observing asymmetrical financial asset returns, i.e., the speed and size of up moves differing from their down moves, [Madan and Wang \(2017\)](#) advocated the use of the BG model, which is the difference of two Gamma processes that allow for different speed parameters. Here, BG is the base univariate distributional model adopted for describing the marginal returns of the multivariate model.

Definition 1.2.7. Let $\Gamma_p(t)$ and $\Gamma_n(t)$ represent two independent Gamma processes with unit mean and variance rates, while $t > 0$ is the time variable. Then, the BG process

$X_{BG}(t)$ is defined by

$$X_{BG}(t) = b_p \Gamma_p(c_p t) - b_n \Gamma_n(c_n t),$$

where parameters b_p, c_p, b_n, c_n can be any positive numbers that represent the scale and speed of positive and negative moves, respectively.

The $X_{BG}(t)$ is not only a Lévy process of independent and identically distributed (i.i.d.) increments but also an infinitely divisible pure jump process with the following characteristic function:

$$\varphi(u) = \left(\frac{1}{1 - i u b_p} \right)^{c_p t} \left(\frac{1}{1 + i u b_n} \right)^{c_n t}, \quad u \in \mathbb{R}. \quad (1.1)$$

Lemma 1.2.1.

(1) Suppose $X \sim BG(b_p, c_{p1}; b_n, c_{n1})$ and $Y \sim BG(b_p, c_{p2}; b_n, c_{n2})$, where X and Y are independent. Then $X + Y \sim BG(b_p, c_{p1} + c_{p2}; b_n, c_{n1} + c_{n2})$.

(2) For $X \sim BG(b_p, c_p; b_n, c_n)$ and $a > 0$, it holds $aX \sim BG(ab_p, c_p; ab_n, c_n)$.

Proof. The asserted properties follow from Expression (1.1) of the characteristic function. □

From the characteristic function (1.1), the BG distributions are stable and are infinitely divisible. The Lévy measure is given by:

$$v(dx) = \left(\frac{c_p}{x} e^{-\frac{x}{b_p}} \mathbb{I}_{(0, \infty)}(x) + \frac{c_n}{|x|} e^{-\frac{|x|}{b_n}} \mathbb{I}_{(-\infty, 0)}(x) \right) dx.$$

Thus, we can also express the characteristic function φ as:

$$\varphi(u) = \exp \left(\int_{\mathbb{R}} (e^{iux} - 1) \frac{k(x)}{x} dx \right), \quad u \in \mathbb{R}.$$

where the specific form for the Lévy density is:

$$k(x) = c_p e^{-\frac{x}{b_p}} \mathbb{I}_{(0, \infty)}(x) - c_n e^{-\frac{|x|}{b_p}} \mathbb{I}_{(-\infty, 0)}(x), \quad x \in \mathbb{R}.$$

In particular, for a $BG(b_p, c_p; b_n, c_n)$ -distributed random variable X , we can specify

- The expectation

$$E[X] := \kappa_1 = b_p c_p - b_n c_n.$$

- The variance

$$\text{Var}[X] = \kappa_2 = b_p^2 c_p + b_n^2 c_n.$$

- The skewness

$$\gamma_1(X) := \frac{\kappa_3}{\kappa_2^{3/2}} = \frac{2(b_p^3 c_p - b_n^3 c_n)}{(b_p^2 c_p + b_n^2 c_n)^{3/2}}.$$

- The kurtosis

$$\gamma_2(X) := 3 + \frac{\kappa_4}{\kappa_2^2} = 3 + \frac{6(b_p^4 c_p + b_n^4 c_n)}{(b_p^2 c_p + b_n^2 c_n)^2}.$$

We can see that the BG distributions are leptokurtic ($\gamma_2(X) > 3$).

For the estimation of the parameters b_p, c_p, b_n, c_n , we first start with the MOM and the MOM estimator can be served as the initial estimator in our estimation procedure. We

define $m_k = E[X_1^k]$ for $k = 1, \dots, 4$ as follows:

$$\hat{m}_k = \frac{1}{n} \sum_{i=1}^n X_i^k. \quad (1.2)$$

From (4.3) in [Küchler and Tappe \(2008\)](#), we can get the following equations,

$$\left\{ \begin{array}{l} \frac{c_p}{b_n} - \frac{c_n}{b_p} - \frac{c_1}{b_p b_n} = 0 \\ \frac{c_p}{b_n^2} + \frac{c_n}{b_p^2} - \frac{c_2}{b_p^2 b_n^2} = 0 \\ \frac{c_p}{b_n^3} - \frac{c_n}{b_p^3} - \frac{c_3}{b_p^3 b_n^3} = 0 \\ \frac{c_p}{b_n^4} + \frac{c_n}{b_p^4} - \frac{c_4}{b_p^4 b_n^4} = 0 \end{array} \right., \quad (1.3)$$

where the constants c_1, c_2, c_3, c_4 defined with $m_1, \dots, m_4$ in Equation (1.2) are given by:

$$\left\{ \begin{array}{l} c_1 = m_1 \\ c_2 = m_2 - m_1^2 \\ c_3 = \frac{1}{2}m_3 - \frac{3}{2}m_1 m_2 + m_1^3 \\ c_4 = \frac{1}{6}m_4 - \frac{2}{3}m_3 m_1 - \frac{1}{2}m_2^2 + 2m_2 m_1^2 - m_1^4 \end{array} \right. \quad (1.4)$$

The density can be expressed in terms of the Whittaker W function. For $x > 0$, the density is given by:

$$\begin{aligned} f_{\theta_0}(x) &= \frac{(b_p)^{-c_p} (b_n)^{-c_n}}{\left(\frac{1}{b_p} + \frac{1}{b_n}\right)^{\frac{1}{2}(c_p+c_n)} \Gamma(c_p)} x^{\frac{1}{2}(c_p+c_n)-1} e^{-\frac{x}{2}\left(\frac{1}{b_p} + \frac{1}{b_n}\right)} \\ &\quad \times W_{\frac{1}{2}(c_p-c_n), \frac{1}{2}(c_p+c_n-1)} \left(x \left(\frac{1}{b_p} + \frac{1}{b_n} \right) \right). \end{aligned} \quad (1.5)$$

For $x < 0$ the roles of b_p, c_p and b_n, c_n are reversed. Here the Whittaker- W function has

the following representation:

$$W_{\lambda,\mu}(z) = \frac{z^\lambda e^{-\frac{z}{2}}}{\Gamma(\mu - \lambda + \frac{1}{2})} \int_0^\infty t^{\mu-\lambda-\frac{1}{2}} e^{-t} \left(1 + \frac{t}{z}\right)^{\mu+\lambda-\frac{1}{2}} dt \quad \text{for } \mu - \lambda > -\frac{1}{2}.$$

We can use the MOM estimator obtained from Equations (1.3) and (1.4) as the initial estimator and then use the density function shown in Equation (1.5) to arrive at the maximum likelihood estimator. However, since we do not have a closed-form density function and the density function of BG has integral representation, the estimation procedure is time-consuming. Therefore, to effectively calculate its density function, we can use the fast Fourier transform (FFT) (Walker, 2017). Using Fourier inversion of the characteristic function (1.1), we can calculate the density function of $X_{BG}(t)$. Then, parameters are obtained by applying maximum likelihood estimation. The Fourier inversion theorem is explained in Definition 1.2.8.

Definition 1.2.8 (Fourier Inversion Formula). If $\varphi_X(\theta)$ is the characteristic function of X and $\int_{\mathbb{R}} |\varphi_X(\theta)| d\theta < \infty$ then X is absolutely continuous with density

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-i\theta x} \varphi_X(\theta) d\theta.$$

1.2.4 Full Gaussian copula model

Introduced by Sklar (1959), the copula is used as a method to construct the joint distribution from several marginal distributions such that various general types of dependence can be represented. The advantage of using the copula is that it allows the separation of the dependence structure from the given marginal distributions. This separation

ration enables one to construct a multivariate distribution with any prespecified marginal distribution and dependence structure. One of the most popular copulas applied in finance is the FGC, which is the copula implied by the multivariate Gaussian distribution. Despite its tractability and simplicity, the main drawback of the FGC is that it fails to capture the tail dependence of financial asset returns, and is known as “The Formula That Killed Wall Street” ([MacKenzie and Spears, 2014](#)) in the 2008 global financial crisis. To compare model performance in explaining multi-asset returns with our MBG model, we briefly review the FGC model, which follows the idea of [Malevergne and Sornette \(2003\)](#).

Following [Nelsen \(2007\)](#) and [Malevergne and Sornette \(2003\)](#), we provide below the definition of the copula of n random variables:

Definition 1.2.9 (Copula). A function $C(\cdot) : [0, 1]^n \rightarrow [0, 1]$ is an n -copula if it has the following properties:

- $\forall u \in [0, 1], C(1, \dots, 1, u, 1, \dots, 1) = u,$
- $\forall u_i \in [0, 1], C(u_1, \dots, u_n) = 0$ if at least one of the u_i equals zero,
- $C(\cdot)$ is grounded and n -increasing, i.e., the C -volume of every box whose vertices lie in $[0, 1]^n$ is positive.

From the definition, the copula is a joint cumulative distribution with support in the unit cube $[0, 1]^n$ and its marginal distributions are uniform distributions. In order for copulas to represent multivariate distributions with arbitrary marginals other than uniform marginal, we need to use Sklar’s theorem ([Sklar, 1959](#)).

Lemma 1.2.2 (Sklar's theorem). For all n -dimensional distribution function F with continuous marginal distribution functions $F_1, \dots, F_n$, there exists a unique n -copula $C : [0, 1]^n \rightarrow [0, 1]$ function such that :

$$F(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n)). \quad (1.6)$$

This theorem can describe multivariate distributions in terms of a copula. Moreover, it is used as a construction schema for copula. Given a multivariate distribution F with marginals $F_1, \dots, F_n$, the function

$$C(u_1, \dots, u_n) = F(F_1^{-1}(u_1), \dots, F_n^{-1}(u_n)) \quad (1.7)$$

is an n -copula corresponding to the multivariate distribution F . This method would be used later in this section to derive copulas such as the Gaussian copula.

If all marginal distribution function F_i is differentiable, and the copula $C(\cdot)$ is n times differentiable, the joint probability density function f could be written as the product of the copula density function $c(\cdot)$ and marginal density functions $f_i(x_i)$:

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= f_1(x_1) \times f_2(x_2) \times \dots \times f_n(x_n) \\ &\times c(F_1(x_1), F_2(x_2), \dots, F_n(x_n)) \end{aligned} \quad (1.8)$$

and the corresponding copula density function $c(\cdot)$ is given by the n -th mixed derivative of the copula function $C(\cdot)$ with respect to each marginal:

$$c(u_1, u_2, \dots, u_n) = \frac{\partial^n C(u_1, u_2, \dots, u_n)}{\partial u_1 \partial u_2 \dots \partial u_n}. \quad (1.9)$$

From Equation (1.8), we can see that a multivariate distribution could be decomposed into marginal probability density functions and copula density functions. Therefore, copula allows for high flexibility in the modeling of multivariate distribution, thereby providing the flexibility for modeling each component's marginal distribution independently. Moreover, because the choice of copula is independent of the marginal distributions, one could select different copula to model different dependence structures. Among the different copulas, one of the most commonly used copulas in the financial industry is the Gaussian copula defined below.

Definition 1.2.10 (Gaussian copula). The Gaussian copula is the copula derived from the multivariate Gaussian distribution. Let Φ denote the cumulative distribution of the standard normal, and $\Phi_{\rho,n}$ be the n -dimensional Gaussian distribution with correlation matrix ρ . Then, the Gaussian n -copula with correlation matrix ρ is

$$C_\rho(u_1, \dots, u_n) = \Phi_{\rho,n}(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_n)) \quad (1.10)$$

and the copula density function is

$$c_\rho(u_1, \dots, u_n) = \frac{\partial C_\rho(u_1, \dots, u_n)}{\partial u_1 \dots \partial u_n}, \quad (1.11)$$

which can be written as

$$c_{\rho}(u_1, \dots, u_n) = \frac{1}{\sqrt{\det \rho}} \exp \left(-\frac{1}{2} \begin{pmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_d) \end{pmatrix}^{\top} \cdot (\rho^{-1} - I) \cdot \begin{pmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_d) \end{pmatrix} \right) \quad (1.12)$$

and $\Phi^{-1}(u_k)$ is the inverse cumulative distribution function of a standard normal distribution and I is an n -by- n identity matrix.

For the FGC model, one estimates the marginal BG models for asset returns with parameters $b_{pi}, c_{pi}, b_{ni}, c_{ni}$ for asset $i = 1, \dots, N$. We then invert the characteristic function of the BG distribution using FFT to obtain the cumulative distribution function of the BG distribution. The density may then be used to construct the BG distribution function. We first transform the marginal returns, $x_i, i = 1, \dots, N$, on stock i into the standard normal variables, $z_i, i = 1, \dots, N$ using the following formula,

$$z_i = \Phi^{-1} \left(F_{BG}^i(x_i) \right),$$

where F_{BG}^i is the BG distribution function estimated from the data. The correlation matrix ρ is the sample correlation matrix of the z_i . To obtain the joint distribution of the original data set, we need to convert the estimated correlated normal variables z_i back to x_i

$$x_i = F_i^{-1}(\Phi(z_i))$$

1.3 Multivariate Bilateral Gamma Model and Inference

1.3.1 Properties of the MBG model

The MBG model is comprised of the MVG and independent BG processes. The construction of the MBG model is a natural extension of the MVG model by [Buchmann et al. \(2019\)](#), where a multivariate Lévy process was constructed with a Lévy density and consistent with prespecified VG marginals with different kurtosis levels for each of the components. The MBG model extends that construction to attain consistency with pre-specified BG marginals. The marginal distributions are BG, with parameters $b_{pi}, c_{pi}, b_{ni}, c_{ni} > 0$ for asset $i = 1, \dots, N$. To be specific,

$$X_{MBG} = X_{MVG} + Y, \quad (1.13)$$

where $Y = (Y_1, \dots, Y_N)^\top$, and $Y_1, \dots, Y_N$ are independent BG variables. The MVG is a multivariate normal with mean vector μ , covariance matrix Σ , both of which are scaled by a gamma variate, G , with unit mean and variance ν . The MVG random vector X can be written as:

$$X_{MVG} = \mu G + \sqrt{G} \sqrt{\text{diag}(\Sigma)} Z,$$

where the vector Z has a multivariate normal distribution with a zero mean, unit variance, and a correlation matrix, C .

All other parameters are derived from the parameters of the marginal processes. We

then obtain the drifts and variance as shown below:

$$\mu_i = \frac{b_p^i - b_n^i}{\nu}, \quad \sigma_i^2 = \frac{2b_p^i b_n^i}{\nu} \quad (1.14)$$

and define the covariance matrix by

$$\Sigma = \Delta(\sigma)C\Delta(\sigma), \quad (1.15)$$

where $\Delta(\sigma)$ is the diagonal matrix with diagonal $\sigma = (\sigma_1, \dots, \sigma_N)^\top$.

The true parameter is given by

$$\theta_0 := (\{b_{ni}\}_{i=1}^N, \{b_{pi}\}_{i=1}^N, \{c_{ni}\}_{i=1}^N, \{c_{pi}\}_{i=1}^N, C, \nu) \in \mathbb{R}^{N(N-1)/2+4N+1}, \quad (1.16)$$

where $C \in \mathbb{R}^{N(N-1)/2}$ and $\nu, b_{ni}, b_{pi}, c_{ni}, c_{pi} \in \mathbb{R}$.

The characteristic function of the MBG distribution is given by

$$\begin{aligned} \varphi(u, \theta_0) &:= E \left[\exp \left(iu^\top X_{MBG} \right) \right] \\ &= \left(\frac{1}{1 - iu^\top \mu \nu + \frac{\nu}{2} u^\top \Sigma u} \right)^{\frac{1}{\nu}} \prod_{i=1}^N \left(\frac{1}{1 - iu_i b_p^i} \right)^{c_p^i - \frac{1}{\nu}} \left(\frac{1}{1 + iu_i b_n^i} \right)^{c_n^i - \frac{1}{\nu}}, \end{aligned} \quad (1.17)$$

where the lower bound of parameter ν is the reciprocal of the minimum of all of the marginal speed parameters $\min_i(\min(c_p^i, c_n^i))$, and u is a column vector.

Proof. Recall from Definition 1.2.4 that the CF of the i -th MVG marginal is defined as

$$\phi_{VG}(u_i; \sigma_i, \nu, \theta_i) = \left(1 - iu_i \theta_i \nu + \frac{1}{2} \sigma_i^2 \nu u_i^2 \right)^{-1/\nu}. \quad (1.18)$$

We consider two independent Gamma distributions, X and Y , with speed and scale parameters as:

$$X \sim \Gamma\left(\frac{1}{\nu}, \sigma_i \alpha_i \sqrt{\nu/2}\right), \quad Y \sim \Gamma\left(\frac{1}{\nu}, \sigma_i \beta_i \sqrt{\nu/2}\right) \quad (1.19)$$

We construct the VG distribution as $Z = X - Y$, where $Z \sim VG(\frac{1}{\nu}, \sigma_i \alpha_i \sqrt{\nu/2}, \sigma_i \beta_i \sqrt{\nu/2})$.

In order to derive the characteristic function of Z , we start by writing the characteristic function of X and Y

$$\phi_X(u_i) = \left(1 - i u_i \sigma_i \alpha_i \sqrt{\nu/2}\right)^{-\frac{1}{\nu}} \text{ and } \phi_Y(u_i) = \left(1 - i u_i \sigma_i \beta_i \sqrt{\nu/2}\right)^{-\frac{1}{\nu}}. \quad (1.20)$$

which lead to

$$\phi_{-Y}(u_i) = \left(1 + i u_i \sigma_i \beta_i \sqrt{\nu/2}\right)^{-\frac{1}{\nu}}.$$

We use the convolution property to sum the two independent random variables X and $-Y$ leading to

$$\begin{aligned} \phi_Z(u) &= \left(1 - i u_i \sigma_i \alpha_i \sqrt{\nu/2}\right)^{-\frac{1}{\nu}} \left(1 + i u_i \sigma_i \beta_i \sqrt{\nu/2}\right)^{-\frac{1}{\nu}} \\ &= \left(1 - i u_i \sigma_i \sqrt{\nu/2} (\alpha_i - \beta_i) + \frac{u_i^2 \sigma_i^2 \nu}{2} \alpha_i \beta_i\right)^{-\frac{1}{\nu}}. \end{aligned} \quad (1.21)$$

Let

$$\theta_i = \frac{\sigma_i (\alpha_i - \beta_i)}{\sqrt{2\nu}}, \quad \text{and} \quad \alpha_i \beta_i = 1. \quad (1.22)$$

Now we identify Equation (1.21) with the CF of marginal VG distribution defined in Equation (1.18). From the construction of MBG introduced in Equation (1.13), we mul-

tiply by the independent BG marginal and get

$$\left(1 - iu_i\sigma_i\alpha_i\sqrt{\nu/2}\right)^{-\frac{1}{\nu}} \left(1 + iu_i\sigma_i\beta_i\sqrt{\nu/2}\right)^{-\frac{1}{\nu}} (1 - iu_ib_p^i)^{-c_p^i} (1 - iu_ib_n^i)^{-c_n^i} \quad (1.23)$$

Let

$$\alpha_i = \sqrt{\frac{2}{\nu}} \frac{b_p^i}{\sigma_i} \quad \text{and} \quad \beta_i = \sqrt{\frac{2}{\nu}} \frac{b_n^i}{\sigma_i}. \quad (1.24)$$

The i -th marginal of BG can be constructed as follows:

$$\begin{aligned} \phi_{BG}(u_i) &= (1 - iu_ib_p^i)^{-(c_p^i + \frac{1}{\nu})} (1 - iu_ib_n^i)^{-(c_n^i + \frac{1}{\nu})} \\ &= (1 - iu_ib_p^i)^{\tilde{c}_p^i} (1 - iu_ib_n^i)^{\tilde{c}_n^i}. \end{aligned} \quad (1.25)$$

where, $\tilde{c}_p^i := c_p^i + \frac{1}{\nu}$ and $\tilde{c}_n^i := c_n^i + \frac{1}{\nu}$.

From Equations (1.22) and (1.24), we can derive

$$\theta_i = \frac{b_p^i - b_n^i}{\nu} \quad \text{and} \quad \sigma_i^2 = \frac{2b_p^i b_n^i}{\nu} \quad (1.26)$$

Therefore, we can form an MBG process with the MVG and independent BG processes.

From the definition of MVG in Definition 1.2.6, we can rewrite the characteristic function of MBG as equation (1.17). □

The multivariate arrival rates or Lévy density $k(x)$ for the MBG model may be specified as follows:

$$k(x) = m(x) + \sum_{i=1}^n k_i(x_i) \prod_{\substack{l \neq i \\ l=1}}^M \mathbf{1}_{x_l=0}$$

where

$$m(x) = \frac{2 \exp(\mu^\top \Sigma^{-1} x)}{\nu(2\pi)^{n/2} \sqrt{|\Sigma|}} \left(\sqrt{\mu^\top \Sigma^{-1} \mu + \frac{2}{\nu}} \right)^{\frac{n}{2}} \left(\sqrt{x^\top \Sigma^{-1} x} \right)^{-\frac{n}{2}} K_{\frac{n}{2}} \left(\sqrt{\left(\mu^\top \Sigma^{-1} \mu + \frac{2}{\nu} \right) x^\top \Sigma^{-1} x} \right).$$

where $K_{\frac{n}{2}}$ is the modified Bessel function of the second kind of order $n/2$, and,

$$k_i(x_i) = \frac{c_n^i - \frac{1}{\nu}}{|x_i|} \exp\left(-\frac{|x_i|}{b_n^i}\right) \mathbf{1}_{x_i < 0} + \frac{c_p^i - \frac{1}{\nu}}{x_i} \exp\left(-\frac{x_i}{b_p^i}\right) \mathbf{1}_{x_i > 0}.$$

1.3.2 Simulation for the MBG model

Monte Carlo methods are algorithms that use statistical sampling to obtain approximate solutions. Usually, Monte Carlo methods solve problems by simulating the underlying process multiple times and calculating the average value of processes. This method has a wide range of applications in finance, such as pricing complex derivative instruments, simulation of events in risk management, and portfolio evaluation. For example, Monte Carlo simulation is the primary technique employed when pricing path-dependent interest rate derivatives such as collateralized mortgage obligations. Simulation methods usually have the advantage of being straightforward for implementation but may suffer from long computation time. However, with easy access to cheap computing power, more financial problems are solved nowadays using simulation. Therefore, it is essential to derive an effective way of simulating the MBG process. We use the sequential sampling technique to simulate data under the MBG model, and the scheme of the univariate BG process and MBG process are both presented below:

Simulating BG as difference of Gammas

Input: BG parameters b_p, c_p, b_n, c_n ; time spacing $\Delta t_1, \dots, \Delta t_m$ s.t. $\sum_{i=1}^m \Delta t_i = T$.

Initialization: Set $X_0 = 0$.

Loop from $i = 1$ to m :

1. Generate $\Delta\gamma_i^+ \sim \Gamma(\Delta t_i \cdot c_p, b_p), \Delta\gamma_i^- \sim \Gamma(\Delta t_i \cdot c_n, b_n)$.
2. Return $X_{t_i} = X_{t_{i-1}} + \Delta\gamma_i^+ - \Delta\gamma_i^-$.

Sequential sampling of MBG process:

Input: parameters $b_p^j, c_p^j, b_n^j, c_n^j, C, \nu$; time spacing $\Delta t_1, \dots, \Delta t_n$ with $\sum_{i=1}^m t_i = T$

Initialization: Set $X_i(0) = 0$

Loop from $i = 1$ to m :

1. Generate $\Delta G_i^0 \sim \Gamma(\Delta t_i / \nu, \nu)$.
2. Generate $\Delta\gamma_{ij}^+ \sim \Gamma(\Delta t_i \cdot \tilde{c}_p^j, b_p^j), \Delta\gamma_{ij}^- \sim \Gamma(\Delta t_i \cdot \tilde{c}_n^j, b_n^j)$.
3. Generate multivariate normal vector $\Delta W_i \sim N(0, \Sigma)$, where Σ is given in Equations (1.14) and (1.15).
4. Return:

$$X_j(t_{i+1}) = X_j(t_i) + \mu_j \Delta G_i^0 + \sqrt{\Delta G_i^0} \cdot \Delta W_{ij} + \Delta\gamma_{ij}^+ - \Delta\gamma_{ij}^-$$

where, $\tilde{c}_p = c_p - \frac{1}{\nu}, \tilde{c}_n = c_n - \frac{1}{\nu}, i = 1, \dots, m, j = 1, \dots, N$

1.3.3 Estimation procedure for the MBG model

The MBG parameters estimation procedure discussed here is based on the minimum distance between theoretical and empirical characteristic functions (CF). When facing distributions without a closed form density function, the estimation methods based on the CF

provide an excellent alternative to the maximum likelihood approach. For example, according to [Feuerverger and McDunnough \(1981\)](#), such CF-based estimation procedures can be based on applying the GMM procedure to a set of discrete moment conditions. The main advantage of this approach is that we can make the estimator's asymptotic variance arbitrarily close to the Cramer–Rao bound by extending the grid. However, the limitation of such an approach is that the covariance matrix of moment conditions becomes singular when the grid size grows larger than the sample size. When the moment condition matrix is not invertible, the GMM objective function explodes; thus, the efficient GMM estimators cannot be computed. To overcome this limitation, the CGMM, developed by [Carrasco and Florens \(2000\)](#) uses the whole continuum of moment conditions associated with the difference between the theoretical and empirical CF. Moreover, an additional penalty term is developed to stabilize the matrix inverse. The CGMM will be adopted in our estimation procedure for MBG because it is particularly useful for random variables whose density functions do not have a closed form while their CF are known.

Starting from the general structure of the CGMM approach, we propose a constrained estimation procedure that involves the whole continuum of moment conditions.

Our estimation procedure consists of the following four steps:

Steps for MBG estimation:

Step 1: Estimate the parameters b_{pi} , c_{pi} , b_{ni} and c_{ni} , for each marginal BG distribution.

Step 2: Reparametrize the MBG $N \times N$ correlation matrix, C , into an $N(N - 1)/2$ vector g . Therefore, vector g that is unrestricted in R^N .

Step 3: Use the CGMM method to estimate the reparametrized correlation matrix g together with the variance rate ν . Note that there is a lower bound on ν , given by $\underline{\nu} = \left(\min_{i=1, \dots, 10} (c_{pi} \wedge c_{ni}) \right)^{-1}$.

Step 4: Recover the correlation matrix, C , from the vector g to obtain the estimation.

1.3.3.1 The CGMM based on characteristic function

Let $X_1, \dots, X_m$ be i.i.d. random vectors with dimension N , Carrasco and Florens (2000) propose to estimate θ based on the moment functions given by

$$d_j(t, \theta) := e^{it^\top X_j} - \varphi(t, \theta), \quad j = 1, \dots, m, \quad (1.27)$$

where $\varphi(t, \theta)$ is the characteristic function of X . In our case, $\varphi(t, \theta)$ is the joint characteristic function of MBG distribution defined in (1.17).

Let $\Pi(t)$ be a probability density function on $\mathbb{R}^d$. We can define $L^2(\Pi)$ as the Hilbert space of complex valued functions that are square integrable with respect to Π :

$$L^2(\Pi) = \left\{ f : \mathbb{R}^d \rightarrow \mathbf{C}, \int f(t) \overline{f(t)} \Pi(t) dt < \infty \right\} \quad (1.28)$$

where $\overline{f(t)}$ represents the complex conjugate of $f(t)$.

We can define the following inner product on $L^2(\Pi) \times L^2(\Pi)$ with finite measure Π :

$$\langle f, g \rangle = \int f(t) \overline{g(t)} \Pi(t) dt. \quad (1.29)$$

Using the notation defined in Equation (1.29), the efficient CGMM estimator can be obtained by solving the following equation:

$$\hat{\theta} = \arg \min_{\theta} \left\langle K^{-1} \hat{d}_m(\cdot, \theta), \hat{d}_m(\cdot, \theta) \right\rangle \quad (1.30)$$

where $\hat{d}_m(t, \theta) = \frac{1}{m} \sum_{j=1}^m d_j(t, \theta)$, and K is the asymptotic covariance operator associated with the moment conditions. The operator K is an integral operator and satisfies:

$$Kf(t_1) = \int_{-\infty}^{\infty} k(t_1, t_2) f(t_2) \Pi(t_2) dt_2, \text{ for any } f \in L^2(\Pi), \quad (1.31)$$

where $k(t_1, t_2)$ is the kernel of the covariance operator K . When $d_j(t_1, \theta)$ is uncorrelated, $k(t_1, t_2)$ is given by $k(t_1, t_2) = E(d_j(t_1, \theta) \overline{d_j(t_2, \theta)})$.

We can estimate the kernel $k(t_1, t_2)$ with sample size m and the consistent first step estimator $\hat{\theta}^1$. The estimated version of $k(t_1, t_2)$ is given by:

$$k_m(t_1, t_2, \hat{\theta}^1) = \frac{1}{m} \sum_{j=1}^m d_j(t_1, \hat{\theta}^1) \overline{d_j(t_2, \hat{\theta}^1)}. \quad (1.32)$$

However, the empirical covariance operator

$$K_m f(t_1) = \int_{-\infty}^{\infty} k_m(t_1, t_2, \hat{\theta}^1) f(t_2) \Pi(t_2) dt_2, \text{ for any } f \in L^2(\Pi)$$

with kernel function given by Equation (1.32) is degenerate and noninvertible. This problem is worsened by the fact that the inverse of K , which K_m is used to estimate, exists only on a dense subset of $L^2(\Pi)$. Moreover, when $g \equiv K^{-1}f$ exists for a given function f , a small perturbation in f may give rise to a large variation in g .

To address these difficulties, we could estimate K^{-1} alternatively:

$$K_{\alpha m}^{-1} = (K_m^2 + \alpha I)^{-1} K_m, \quad (1.33)$$

where $K_{\alpha m}^{-1}$ is a regularized inverse of K_m .

Similar to the GMM estimator in Hansen (1982), the CGMM estimator can also be implemented in two steps. In the first step, one estimates K by a sample analog K_m based on a consistent first step estimator. A consistent estimator $\hat{\theta}_1^*$ of θ can be obtained by minimizing the right-hand side of Equation (1.34):

$$\hat{\theta}_1^* = \operatorname{argmin}_{\theta \in \Theta} \int \hat{d}_m(t, \theta) \overline{\hat{d}_m(t, \theta)} \Pi(t) dt, \quad (1.34)$$

where $\hat{d}_m(t, \theta) = \frac{1}{m} \sum_{j=1}^m d_j(t, \theta)$. The $\overline{\hat{d}_m(t, \theta)}$ represents the conjugate of $\hat{d}_m(t, \theta)$, and Π is an arbitrary finite measure.

Therefore, from Equation (1.30), the feasible second step CGMM estimator is given

by:

$$\hat{\theta}(\alpha) = \arg \min_{\theta} \left\langle K_{\alpha m}^{-1} \hat{d}_m(\cdot, \theta), \hat{d}_m(\cdot, \theta) \right\rangle. \quad (1.35)$$

We use the minimization problem in Equation (1.37) for the CGMM with penalty term estimation. Note that we choose $\Pi(\cdot)$ to be a multivariate Gaussian density since this particular choice improves the accuracy of numerical integration.

Under mild conditions in Theorem 1 by Carrasco and Kotchoni (2017), we have the consistency and asymptotic normality for the two-step CGMM estimator.

Theorem 1.3.1. Suppose the i.i.d. data $\{X_i\}_{i=1}^m \sim X \in \mathbb{R}^N$ with a density function $f_{\theta_0}(x)$ without closed form. The true parameter $\theta_0 \in \mathbb{R}^{4N+N(N-1)/2+1}$ is defined by (1.16), which is an interior point of a compact parameter space Θ . We assume that

$$Ed_1(t, \theta), \frac{\partial \varphi(t, \theta)}{\partial \theta_j}, \text{ and } \frac{\partial^2 \varphi(t, \theta)}{\partial \theta_j \partial \theta_k} \in \{f \in L^2(\pi) \text{ such that } \|K^{-\lambda} f\| < \infty\} \quad (1.36)$$

for all $\theta \in \Theta$ and for some $\lambda \geq 1$ (e.g. $\lambda = 1$), where $d_1(t, \theta)$ is given by (1.27) and $\varphi(t, \theta)$ is defined in (1.17). The CGMM estimator $\hat{\theta}_T(\alpha)$ in (1.35) is consistent and asymptotically normal:

$$m^{1/2}(\hat{\theta}(\alpha) - \theta_0) \xrightarrow{L} N(0, I_{\theta_0}^{-1}),$$

as m and $\alpha m^{1/2}$ go to infinity and α goes to zero, where $I_{\theta_0}^{-1}$ denotes the inverse of the Fisher information matrix defined by $I_{\theta_0} := E[\nabla \log f_{\theta_0}(X) \nabla \log f_{\theta_0}(X)^\top]$.

In practice, we use the Carrasco et al. (2007) Proposition 3.4 as our objective function.

Lemma 1.3.1 (Carrasco et al. (2007)). In practice, we solve Equation (1.37) to obtain $\hat{\theta}$, which is equivalent as solve Equation (1.35),

$$\hat{\theta}^*(\alpha) = \operatorname{argmin}_{\theta \in \Theta} \left(\overline{v(\theta)}^\top [\alpha I + C^2]^{-1} v(\theta) \right), \quad (1.37)$$

where $v(\theta)$ is an m -dimensional vector and its j -th entry is defined as:

$$v_j(\theta) = \int \overline{d_j(t, \hat{\theta}_1^*)} \hat{d}_m(t, \theta) \Pi(t) dt \quad (1.38)$$

and we define an $m \times m$ matrix C where entry c_{lj} is:

$$c_{lj} = \frac{1}{m - q} \int \overline{d_l(t, \hat{\theta}_1^*)} d_j(t, \hat{\theta}_1^*) \Pi(t) dt, \quad (1.39)$$

where q is the number of dimension of parameters.

See Appendix 1.6 for proof.

1.3.3.2 Reparametrization of correlation matrix in the MBG model

The naive approach for estimating the dependency parameters for the MBG model in three or more-dimensional cases is to apply the CGMM algorithm on the correlation matrix C directly. However, the critical issue that would arise is that we cannot guarantee that the resulting correlation matrix C that we try to estimate is a positive semidefinite matrix in each iteration of the algorithm. To address this issue, we seek a method that reparametrizes the correlation matrix as a vector that is unrestricted in $\mathbb{R}^N$. The literature

has proposed several methods for performing such decomposition on covariance matrices, including Cholesky decomposition and the spherical trigonometric transformation. However, in the construction of the MBG, the variances of the multivariate distribution have been fixed by its marginal distributions. Therefore, instead of the covariance matrix, the correlation matrix is the target that needs to be reparametrized. Archakov and Hansen (2021) proposed a transformation of the lower diagonal of the matrix logarithm of the correlation matrix, and this also establishes a one-to-one relationship between the set of $N \times N$ non-singular correlation matrices and vectors in $\mathbb{R}^{N(N-1)/2}$.

Definition 1.3.1 (New Parametrization of Correlation Matrices). For a nonsingular correlation matrix, C , let function g be the following parametrization:

$$g(C) := \text{vecl}(\log C),$$

where $\log(C)$ is the matrix logarithm of matrix, C , and vecl is the operator that vectorizes the lower off-diagonal elements of a square matrix.

To illustrate this mapping, consider the case with a 3×3 correlation matrix, C . We have

$$G_c = \log C = \begin{pmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{pmatrix} \quad (1.40)$$

$$g = \text{vecl}(\log C) = \begin{pmatrix} G_{21} \\ G_{31} \\ G_{32} \end{pmatrix} \quad (1.41)$$

where G_c denotes the logarithm transformed correlation matrix, and g denotes the resulting vector after apply the vecl operator on $\log C$. We can see from the above example that the diagonal elements of $\log C$ were discarded. The authors argue that one can reconstruct a correlation matrix from an arbitrary vector, g , and the reconstructed correlation matrix is unique.

Lemma 1.3.2 (Archakov and Hansen (2021)). For any real symmetric matrix, $A \in \mathbb{R}^{n \times n}$, there exists a unique vector, $x^* \in \mathbb{R}^n$, such that $\exp(f(A, x^*))$ is a correlation matrix.

Here, $\exp$ stands for the matrix exponential, and $f(A, v)$ stands for the operation that replaces the diagonal elements in matrix A with vector v .

In line with the previous example for the 3×3 correlation matrix, for any vector $g = (g_1, g_2, g_3)^\top$, there exists a vector $\omega(g) = (x_1, x_2, x_3)^\top$, such that

$$\exp \begin{pmatrix} x_1 & g_1 & g_2 \\ g_1 & x_2 & g_3 \\ g_2 & g_3 & x_3 \end{pmatrix}$$

is a valid correlation matrix, and the solution $\omega(g)$ is unique.

Let $G(g, \omega)$ be the symmetric matrix with lower diagonal elements $\text{vecl}(g)$ and diagonal elements ω . From Lemma 1.3.2, for any g , there exists a unique vector $\omega^*(g)$

such that $\exp(G(g, \omega^*(g)))$ is a correlation matrix.

With the following iterative algorithm, with arbitrary initial vector $\omega_{(0)}$, we can calculate $\omega^*(g)$ for any vector g :

$$\omega_{(k+1)} = \omega_{(k)} - \log(\text{diag}(\exp(G(g, \omega_{(k)}))) , \quad k = 0, 1, 2, \dots, \quad (1.42)$$

where *diag* is the operation that extracts the diagonal elements of the matrix and $\log$ is the element-wise logarithm. As $\omega_{(k)} \rightarrow \omega^*$, ω^* is the solution.

Algorithm 1.3.1 Construct correlation matrix C from arbitrary real vector

Input: vector $g \in \mathbb{R}^{N(N-1)/2}$

initial vector $\omega_{(0)} \in \mathbb{R}^N$

tolerance T

max iteration I_{max}

$i \leftarrow 0, \omega_i \leftarrow \omega_{(0)}, \Delta \leftarrow \infty$

while not ($\Delta < T$ **or** $i > I_{max}$) **do**

$\omega_{prev} = \omega_i$

$\omega_i = \omega_i - \log(\text{diag}(\exp(G(g, \omega_i))))$

$\Delta = ||\omega_i - \omega_{prev}||$

$i = i + 1$

end while

$C \leftarrow \exp(G(g, \omega_i))$

Output: $N \times N$ correlation matrix C

For convenience, we define **Algorithm 1.3.1**($g, \omega_{(0)}, T, I_{max}$) as the output of **Algorithm 1.3.1**.

1.3.3.3 Proposed algorithms and simulation studies

Algorithm 1.3.2 Objective Function to Calculate $\hat{\theta}_1^*$

Input: $N; m$ ▷ MBG dimension; Number of the observations
 $b_n^i, b_p^i, c_n^i, c_p^i$ for $i = 1, \dots, N$ ▷ BG marginal parameters
 ν_0 ▷ One of MBG parameters
 $g_0 \in \mathbb{R}^{N(N-1)/2}$ ▷ Initial vector from the input of **Algorithm 1.3.1**
 $X_{m \times N}$ ▷ Asset returns Matrix of $m \times N$
 S ▷ Sample size for applying Monte Carlo integration
 $B_{S \times N}$ ▷ Matrix of $S \times N$ for pre generated Monte Carlo Samples

$C \leftarrow \text{Algorithm 1.3.1}(g_0, \omega_{(0)}, T, I_{max})$ ▷ to calculate MBG Correlation Matrix C
 $\Phi_{emp} \leftarrow \exp(iX_{m \times N}B_{S \times N}^\top); A \leftarrow \mathbf{1}$ ▷ Matrix of $m \times S$; all one matrix of $m \times 1$
 $\phi_{theo} \leftarrow \phi(C, \nu, b_n, b_p, c_n, c_p, B_{S \times N})$ ▷ evaluate MBG CF for each row of $B_{S \times N}$
 $\Phi_{theo} \leftarrow A\phi_{theo}^\top$ ▷ ϕ_{theo} is $S \times 1$
 $d \leftarrow \Phi_{emp} - \Phi_{theo}$ ▷ d is $m \times S$
 $\hat{d}_m \leftarrow \frac{1}{M}d^\top A$ ▷ $\hat{d}_m$ is vector of $S \times 1$
 $z \leftarrow \frac{1}{m}\hat{d}_m^\top \hat{d}_m$

Output: z ▷ the value of the objective function that need to minimize in (1.34)

Following **Algorithm 1.3.2** to minimize the objective function to calculate

$$\hat{\theta}_1^* = \{\nu_1^*; g_1^*\},$$

where $\hat{\theta}_1^*$ is the parameter vector that concatenates scalar ν_1^* and vector g_1^* .

Algorithm 1.3.3 Objective function to calculate $\hat{\theta}_T^*$

Input: $\{b_n^i, b_p^i, c_n^i, c_p^i\}_{i=1}^n, \nu_0, g_0, X_{m \times N}, B_{S \times N}, \nu_1^*, g_1^*,$

$\bar{\alpha}$

▷ hyperparameter for CGMM.

$\Phi_{emp} \leftarrow \exp(iX_{m \times N}B_{S \times N}^\top), A \leftarrow \mathbf{1}$

$C_{\theta_1} \leftarrow \text{Algorithm 1.3.1}(g_1^*, \omega_{(0)}, T, I_{max})$ ▷ to calculate MBG Correlation Matrix C

$\phi_{\theta_1, theo} \leftarrow \phi(C_{\theta_1}, \nu_1^*, b_n, b_p, c_n, c_p, B_{S \times N})$ ▷ evaluate MBG CF for each row of $B_{S \times N}$

$\Phi_{\theta_1, theo} \leftarrow A\phi_{\theta_1, theo}^\top$ ▷ $\phi_{\theta_1, theo}$ is $S \times 1$

$d_{\theta_1} \leftarrow \Phi_{emp} - \Phi_{\theta_1, theo}$ ▷ d_{θ_1} is $m \times S$

$C_\theta \leftarrow \text{Algorithm 1.3.1}(g_0^*, \omega_{(0)}, T, I_{max})$ ▷ to calculate MBG Correlation Matrix C_θ

$\phi_{theo} \leftarrow \phi(C_\theta, \nu_0, b_n, b_p, c_n, c_p, T)$ ▷ evaluate MBG CF for each row of T

$\Phi_{theo} \leftarrow A\phi_{theo}^\top$ ▷ ϕ_{theo} is $S \times 1$

$d \leftarrow \Phi_{emp} - \Phi_{theo}, \hat{d}_m \leftarrow \frac{1}{M}d^\top A$

$v \leftarrow \frac{1}{m}\overline{d_{\theta_1}}\hat{d}_m$ ▷ v is vector of $m \times 1$

$C \leftarrow \frac{1}{M - \text{card}(\{\nu_0; g_0\})} \frac{1}{m}\overline{d_{\theta_1}}d_{\theta_1}^\top$ ▷ C is matrix of $m \times m$

$z \leftarrow \bar{v}^\top [\bar{\alpha}I + C^2]^{-1}v$

Output: z_α ▷ value of the objective function that was to be minimized in (1.37)

Upon retrieving the result $\hat{\theta}_1^*$ from Algorithm 1.3.2, we first unpack $\hat{\theta}_1^*$ into ν_1^* and g_1^* and set up the objective function to calculate $\hat{\theta}^*$ using Algorithm 1.3.3. With the objective function, we then plug it into any optimization routine to minimize the object function to calculate

$$\hat{\theta}_T^* = \{\nu^*; g^*\}$$

and $\hat{\theta}_T^*$ is our CGMM estimator.

To verify the accuracy of the CGMM method in three- and four-dimensional cases with simulated data, we use a Monte Carlo simulation to simulate 100 batches of samples

with a sample size of 3,000 from the MBG model. After simulation, 100 estimates are obtained using the CGMM estimation method for each batch of the 3,000 data points. Sample standard error (sd.), mean, and other statistics are calculated using bootstrapped parameters. Estimated parameters for three- and four-dimensional cases are presented in Tables 1.1 and 1.2, respectively.

Table 1. CGMM estimation (3-dimension)								
	True	Est.	sd.	min	25%	50%	75%	max
$C_{1,2}$	0.7	0.693	0.084	0.510	0.656	0.695	0.748	0.861
$C_{1,3}$	0.5	0.500	0.061	0.341	0.471	0.510	0.525	0.611
$C_{2,3}$	0.2	0.206	0.058	0.122	0.156	0.209	0.246	0.313
ν	2	1.986	0.190	1.610	1.878	2.028	2.137	2.263

Table 1.1: CGMM 3-dimension estimation result.

Table 2. CGMM estimation (4-dimension)								
	True	Est.	sd.	min	25%	50%	75%	max
$C_{1,2}$	0.4	0.404	0.177	-0.064	0.275	0.393	0.500	0.920
$C_{1,3}$	0.7	0.722	0.112	0.508	0.619	0.734	0.939	0.989
$C_{1,4}$	-0.4	-0.462	0.172	-0.925	-0.621	-0.445	-0.275	-0.216
$C_{2,3}$	0.3	0.372	0.096	0.149	0.300	0.368	0.449	0.585
$C_{2,4}$	0.25	0.032	0.180	-0.265	-0.093	-0.036	0.161	0.438
$C_{3,4}$	-0.6	-0.607	0.120	-0.798	-0.682	-0.624	-0.559	-0.142
ν	2	2.262	0.216	1.720	2.158	2.303	2.409	2.595

Table 1.2: CGMM 4-dimension estimation result.

From Tables 1.1 and 1.2, we can see that the mean and the median of the bootstrapped parameters are both close to the true value of the parameters and the standard deviations are relatively small.

1.4 Performance on Fitting Stock Returns

To test the performance of the MBG model, we use the FGC model as our benchmark and the performance metric introduced by Madan (2020) that can be used to evaluate model performance.

We use the Anderson–Darling distance introduced by Anderson and Darling (1952) and used by Madan (2020) as our performance metric. The performance metric adopted here is used to compare the similarity between the observations sampled from our model

and the actual data. This metric puts more weight on extreme observations with a lower bound of zero and no upper bound. A smaller value indicates that we are more confident that our samples are drawn from the empirical distribution of the actual data.

After the model has been successfully estimated from observations, we sample a batch of data from the fitted distribution with the same size as the observation. For N -dimensional observations of size m , we use x_{ik}, y_{ik} to denote the k -th dimension of i -th observation and k -th dimension of i -th sampled data, respectively. For the i -th observation, we use T_i to denote the proportions of data in our observations that are more extreme than those in the i -th observation.

$$T_i = \frac{1}{m} \text{card}(C_i) \quad (1.43)$$

where

$$C_i = \{x_t \mid \text{sign}(x_{ik})(x_{tk} - x_{ik}) > 0, \forall t \in \{1, \dots, m\}, k \in \{1, \dots, N\}\} \quad (1.44)$$

is the set containing observations with more extreme values in each dimension than the i -th observation. Similarly, we define $\hat{T}_i$ as the proportions of data in our simulated sample that are more extreme than those in the i -th observation. Moreover, $\hat{C}_i$ is the set containing sample data that have more extreme values in each dimension compared to the i -th observation.

$$\hat{T}_i = \frac{1}{m} \text{card}(\hat{C}_i), \quad (1.45)$$

where

$$\hat{C}_i = \{y_t \mid \text{sign}(x_{ik})(y_{tk} - x_{ik}) > 0, \forall t \in \{1, \dots, m\}, k \in \{1, \dots, N\}\}. \quad (1.46)$$

With T_i calculated from actual observation, and $\hat{T}_i$ calculated from simulated data, the performance metric Z is:

$$Z = \sum_{i=1}^m \frac{(T_i - \hat{T}_i)^2}{T_i(1 - T_i)}. \quad (1.47)$$

In the above equation, each $(T_i - \hat{T}_i)^2$ measures the squared distance between the proportion calculated from the simulated data and the proportion calculated from actual observations. The term, $T_i(1 - T_i)$, could be viewed as the weight that we apply to the i -th squared distance. For the i -th observation, the fewer number of times we record a more extreme observation (tail event), the smaller the T_i would be, therefore the bigger the weight $\frac{1}{T_i(1-T_i)}$ would become. The weight $\frac{1}{T_i(1-T_i)} \rightarrow \infty$ as $T_i \rightarrow 0$ or $T_i \rightarrow 1$.

The value of the metric Z is bounded below by 0. The smaller the value of Z , the more confident we are that the simulated data are drawn from the actual data's empirical distribution function. Therefore, a smaller Z indicates a better fit for the model.

From the two datasets—the actual returns data and the simulated data with calibrated parameters in the model—we first use the actual returns data to calculate T_i defined using Equations (1.43) and (1.44).

To illustrate the method mentioned above, we compare the performance of the FGC and MBG models in actual stock returns data of 28 stocks in the Dow Jones index for the 2010–2020 period. The results are presented in Table 1.3.

Table 1.3. MBG & FGC comparison using Anderson-Darling distance							
Tickers	FGC	MBG	Diff	Tickers	FGC	MBG	Diff
AAPL_INTC_MSFT	2.68	0.08	2.6	HD_NKE_RTX	2.2	0.07	2.13
AXP_BA_RTX	2.84	0.12	2.72	HD_RTX_UNH	2.17	0.07	2.1
AXP_DIS_JPM	2.69	0.09	2.61	IBM_MMM_RTX	2.53	0.11	2.42
AXP_GS_JPM	2.24	0.21	2.03	JNJ_KO_PG	2.11	0.13	1.98
AXP_HD_TRV	2.33	0.08	2.25	JNJ_MMM_WBA	1.97	0.15	1.82
AXP_RTX_V	2.82	0.1	2.72	JNJ_MRK_PFE	1.89	0.12	1.77
CAT_MMM_RTX	1.97	0.07	1.9	JNJ_PFE_PG	1.82	0.11	1.71
CAT_MMM_XOM	1.81	0.06	1.75	JNJ_PG_WMT	1.84	0.1	1.74
CSCO_INTC_MSFT	2.4	0.07	2.33	JPM_RTX_TRV	2.82	0.1	2.72
CVX_JPM_XOM	2.29	0.17	2.12	KO_PG_TRV	1.69	0.18	1.51
HD_MCD_TRV	2.09	0.07	2.02	KO_PG_VZ	1.99	0.07	1.92

Table 1.3: MBG and FGC comparison using Anderson-Darling distance

From Table 1.3, we can see that the MBG model outperforms the FGC model in terms of the value of Anderson-Darling distance z .

1.5 Conclusion and Future Research

This chapter discusses a multivariate Lévy process called the MBG process. This process is a natural extension of the MVG process with prespecified BG marginal. The MBG process is most suited for modeling multivariate financial assets returns since it nicely captures the negative skewness, excess kurtosis, the asymmetry in the upward and downward motions, and the tail dependence structure of asset returns. Considering the construction

of the new multivariate process, we propose a Monte Carlo simulation procedure for effectively sampling data under this model. After discussing the existing estimation methods in the literature, we propose an estimation procedure for calibrating the MBG process that minimizes the distance between the empirical and theoretical CF. The estimation steps involve first estimating the parameters for each of MBG's marginal BG distribution, then reparametrizing its correlation matrix into a vector that is unrestricted in R^N , applying the CGMM method to estimate the reparametrized correlation matrix together with the variance rate, and finally recovering the correlation matrix to obtain the estimation. A numerical analysis of the simulated data set is conducted to illustrate the proposed estimation procedure. We also test this estimation procedure on real data by comparing them with the FGC model and fitting stock returns in the Dow Jones Index; we conclude that the MBG significantly outperforms FGC. Possible future research could adopt the estimation procedure for distribution that has correlated dependent structure with pre-defined marginal like the MVG model. Moreover, we can compare the performance of t-copula with BG marginal and the MBG model by fitting real stock returns. Another research direction could be to determine alternative test statistics for multivariate distributions, such as the Anderson-Darling tests introduced by [Scholz and Stephens \(1987\)](#), the multivariate Kolmogorov–Smirnov goodness-of-fit test proposed by [Justel et al. \(1997\)](#), and the Wasserstein two-sample test described by [Ramdas et al. \(2017\)](#).

1.6 Appendix

The appendix includes the proofs of the main results.

1.6.1 The Proof of Lemma 1.3.1

Proof. Let $f = K_{\alpha m}^{-1} \hat{d}_m(\theta)$, from Equation (1.33), we have

$$(K_m^2 + \alpha I) f = K_m \hat{d}_m(\theta). \quad (1.48)$$

We define b_l as $l = 1, \dots, m$,

$$b_l = \int \overline{h_l(t, \hat{\theta}_1)} \cdot f(t) \cdot \Pi(t) dt. \quad (1.49)$$

Then from Equations (1.31) and (1.32), we obtain the following equation,

$$K_m^2 f(t) + \alpha f(t) = \frac{1}{m-q} \sum_{j,l=1}^m d_j(t, \hat{\theta}_1) \cdot c_{jl} \cdot b_l + \alpha f(t), \quad (1.50)$$

where matrix, C , is as defined in Equation (1.39).

Similarly, we have,

$$K_m \hat{d}_m(\theta) = \frac{1}{m-q} \sum_{j=1}^m d_j(t, \hat{\theta}_1) \cdot v_j(\theta), \quad (1.51)$$

where $v(\theta)$ is defined in Equation (1.38).

Therefore, we can rewrite Equation (1.48) as shown below:

$$\frac{1}{m-q} \sum_{j,l=1}^m d_j(t, \hat{\theta}_1) \cdot c_{jl} \cdot b_l + \alpha f(t) = \frac{1}{m-q} \sum_{j=1}^m d_j(t, \hat{\theta}_1) \cdot v_j(\theta). \quad (1.52)$$

Multiply Equation (1.52) by $\overline{d_k(t, \hat{\theta}_1)} \Pi(t)$ and integrate it with respect to t , we have

$$\sum_{j,l=1}^m c_{kj} c_{jl} b_l + \alpha b_k = \sum_{j=1}^m c_{kj} v_j(\theta). \quad (1.53)$$

We can rewrite Equation (1.53) by using matrix notation,

$$[\alpha I + C^2] \underline{b} = C \underline{v}(\theta). \quad (1.54)$$

where b is an m -dimensional vector, I is an m -dimension identity matrix, and C is an $m \times m$ matrix and its entry is defined in Equation (1.39).

From Equation (1.54), we can solve vector b as:

$$\underline{b} = [\alpha I + C^2]^{-1} C \underline{v}(\theta). \quad (1.55)$$

Now we multiply Equation (1.52) by $\overline{\hat{d}_m(t; \theta)} \Pi(t)$ and then integrate with respect to t . We have

$$\alpha \langle f, \hat{d}_m(\theta) \rangle + \frac{1}{m-q} \sum_{j,l=1}^m \overline{v_j(\theta)} c_{jl} b_l = \frac{1}{m-q} \sum_{j=1}^m \overline{v_j(\theta)} v_j(\theta). \quad (1.56)$$

Therefore, we have,

$$\langle f, d_m(\theta) \rangle = \frac{1}{\alpha(m-q)} \left[\overline{\underline{v}^\top(\theta)} \underline{v}(\theta) - \overline{\underline{v}^\top(\theta)} C \underline{b} \right]. \quad (1.57)$$

From Equation (1.55), we insert $\underline{b}$ into Equation (1.57) and obtain the following equation:

$$\langle f, d_m(\theta) \rangle = \frac{1}{\alpha(m-q)} \overline{\underline{v}^\top(\theta)} \left[I - C [\alpha I + C^2]^{-1} C \right] \underline{v}(\theta). \quad (1.58)$$

Now we need to show that,

$$I - C [\alpha I + C^2]^{-1} C = \alpha [\alpha I + C^2]^{-1}. \quad (1.59)$$

It is similar to show that:

$$I - \alpha [\alpha I + C^2]^{-1} = C [\alpha I + C^2]^{-1} C. \quad (1.60)$$

Since C is a hermitian matrix, $C = \bar{C}^\top$ and $C = UDU^{-1}$, where D is a diagonal matrix.

Thus, we have

$$\begin{aligned}
I - \alpha [\alpha I + C^2]^{-1} &= I - \alpha U [\alpha I + D^2]^{-1} U^{-1} \\
&= U \left[U^{-1} U - \alpha [\alpha I + D^2]^{-1} \right] U^{-1} \\
&= U \left[I - \alpha [\alpha I + D^2]^{-1} \right] U^{-1} \\
&= U \left[\frac{1}{\alpha} [\alpha I + D^2] - I \right] \alpha [\alpha I + D^2]^{-1} U^{-1} \\
&= U D^2 [\alpha I + D^2]^{-1} U^{-1} \\
&= C [\alpha I + C^2]^{-1} C.
\end{aligned}$$

Inserting Equation (1.59) into (1.58), we have

$$\langle f, d_m(\theta) \rangle = \frac{1}{m - q} \overline{v(\theta)}^\top [\alpha I + C^2]^{-1} \underline{v}(\theta). \quad (1.61)$$

Therefore, from Carrasco et al. (2007), we obtain $\hat{\theta}$, as follows:

$$\hat{\theta}^*(\alpha) = \underset{\theta \in \Theta}{\operatorname{argmin}} \left(\overline{v(\theta)}^\top [\alpha I + C^2]^{-1} \underline{v}(\theta) \right). \quad (1.62)$$

□

1.6.2 The Proof of Theorem 1.3.1

To show the asymptotic efficiency of the CGMM estimators for i.i.d. data setting, we review the following five assumptions:

Let $X_t \in \mathbb{R}^p$ be a random vector whose distribution is indexed by a finite dimen-

sional parameter, θ , with a true value θ_0 . When the process $\{X_t\}_{t \geq 1}$ is i.i.d., we consider the CGMM estimator for θ_0 based on the moment function given by:

$$h_t(\tau, \theta; \theta_0) = e^{i\tau'X_t} - \varphi(\tau, \theta),$$

where $\varphi(\tau, \theta) = E^\theta(e^{i\tau'X_t})$ is the characteristic function of X_t , E^θ is the expectation operator with respect to the data generating process indexed by θ , and θ_0 is the true parameter on which $h_t(\tau, \theta; \theta_0)$ depends implicitly via the actual data X_t .

Assumption 1: The probability density function $\Pi(\cdot)$ is strictly positive on $\mathbb{R}^d$ and admits all its moments.

Assumption 2: For all $t \in \mathbb{R}^d$, the equation $E^{\theta_0}(d_j(t, \theta)) = 0$ has a unique solution θ_0 , where θ_0 is an interior point of a compact set Θ .

Assumption 3: (i) $d_j(t, \theta)$ is three times continuously differentiable with respect to θ . Furthermore, (ii) the first two derivatives of $d_j(t, \theta)$ with respect to its second argument satisfy:

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \frac{\partial d_j(t, \theta)}{\partial \theta_j} - E\left(\frac{\partial d_j(t, \theta)}{\partial \theta_j}\right) &= O_p(T^{-1/2}) \\ \frac{1}{T} \sum_{t=1}^T \frac{\partial^2 d_j(t, \theta)}{\partial \theta_j \partial \theta_k} - E\left(\frac{\partial^2 d_j(t, \theta)}{\partial \theta_j \partial \theta_k}\right) &= O_p(T^{-1/2}), \end{aligned}$$

for all j and k .

Assumption 4: (i) $E^{\theta_0}(d_j(t, \theta)) \in \Phi_\lambda$ for all $\theta \in \Theta$ and for some $\lambda \geq 1$, where

$$\Phi_\lambda := \{f \in L^2(\Pi) \text{ such that } \|K^{-\lambda}f\| < \infty\}.$$

Furthermore, (ii) the first two derivatives of $E^{\theta_0}(d_j(t, \theta))$ with respect to θ belong to Φ_λ

for all θ in a neighborhood of θ_0 and for the same λ as above.

Assumption 5: (i) The random variables $\{X_i\}_{i=1}^m$ are i.i.d. and satisfy $X_i = X(\theta_0, \varepsilon_i)$ where $X(\theta_0, \varepsilon_i)$ is three times continuously differentiable with respect to θ_0 and ε_i is i.i.d. noise whose distribution is known and does not depend on θ_0 . Furthermore, (ii) the gradient, $G(\tau, \theta; \theta^0) = E\left(\frac{\partial h_t(\tau, \theta; \theta^0)}{\partial \theta}\right)$ and covariance operator, K , are continuously differentiable with respect to in θ^0 .

Based on the Assumptions 1–5, one has the following results for the two-step CGMM estimator; see Theorem 1 of Carrasco and Katchoni (2017) or Katchoni et al. (2019).

Proposition 1.6.1. Under Assumptions 1–5, the CGMM estimator $\hat{\theta}_T(\alpha)$ defined in Carrasco and Katchoni (2017) is consistent and asymptotic normality satisfy

$$T^{1/2} \left(\hat{\theta}_T(\alpha) - \theta_0 \right) \xrightarrow{L} N(0, I_{\theta_0}^{-1}),$$

as T and $\alpha T^{1/2}$ goes to infinity and α goes to zero, where $I_{\theta_0}^{-1}$ denotes the inverse of the Fisher information matrix.

The proof of Theorem 1.3.1

The Assumption 1 automatically holds for the multivariate Gaussian density $\Pi(\cdot)$.

For Assumption 2, we have

$$Ed_1(t, \theta) := Ee^{it^\top X_1} - \varphi(t, \theta) = 0, \quad \forall t \in \mathbb{R}^N. \quad (1.63)$$

And θ_0 is an interior point of a compact parameter space Θ .

For Assumption 3(ii), $d_1(t, \theta)$ is three times continuously differentiable with respect to θ . For the i.i.d. data, the derivative of $d_1(t, \theta)$ with respect to θ is constant and nonrandom. Thus, Assumption 3(ii) holds.

The Assumption 4 follows by our assumption

$$Ed_1(t, \theta) \in \{f \in L^2(\pi) \text{ such that } \|K^{-\lambda}f\| < \infty\}$$

for all $\theta \in \Theta$ and for some $\lambda \geq 1$ (e.g. $\lambda = 1$).

For Assumption 5(i), $X(\theta_0, \varepsilon_i)$ is three times continuously differentiable with respect to θ_0 and ε_t is a i.i.d. noise whose distribution is known and does not depend on θ_0 . For Assumption 5(ii), the gradient $G(\tau, \theta; \theta^0) = E\left(\frac{\partial d_1(t, \theta)}{\partial \theta}\right) = E\left(\frac{\partial \varphi(t, \theta)}{\partial \theta}\right)$ and covariance operator K in (1.31) are continuously differentiable with respect to in θ^0 .

Chapter 2: Pairs Trading Strategies: Distance, Cointegration, Copula, and Multivariate Bilateral Gamma Method

2.1 Introduction

Pair trading is categorized as a statistical arbitrage and convergence trading strategy based on past stock prices and their comovement, while maintaining market neutrality. This strategy opens trade when the prices of two stocks diverge from equilibrium and closes trade when their relationship is restored. Wall Street practitioners have been using this strategy for decades since the 1980s, and one of the pioneers is Nunzio Tartaglia, a quant researcher in Morgan Stanley. He assembled a team of mathematicians and computer scientists to develop this equity long–short market neutral strategy using sophisticated statistical methods and automated trading systems. Since then, this strategy has become a popular investment strategy used by institutions and hedge funds ([Gatev et al., 2006](#)).

The most cited paper in this domain, as well as the first literature to research on pair trading strategy, is [Gatev et al. \(2006\)](#). This paper proposes a simple yet efficient method called the distance method. In this strategy, the spread of two scaled time series of prices is monitored as a trading signal and the strategy profits from placing bets on

the reversion of spread when it diverts. This strategy records an annualized excess return of 11%, with low exposure to systematic risks. According to [Krauss \(2017\)](#), there are four streams of approaches discussed in the literature to construct pair trading strategies, besides the distance method. The first is the cointegration method, which models the comovement of stocks as a cointegrated system ([Vidyamurthy, 2004](#); [Lin et al., 2006](#); [Caldeira and Moura, 2013](#)). The second is the time-series approach, which models the spread in state space ([Elliott et al., 2005](#); [Cummins and Bucca, 2012](#)). Another is the stochastic control approach, which models the dynamics using the Ornstein–Uhlenbeck process, or with error correction models ([Jurek and Yang, 2007](#); [Liu and Timmermann, 2013](#)). The last is the copula approach that uses a copula to model the joint dynamic of stocks ([Liew and Wu, 2013](#); [Xie and Wu, 2013](#)). These existing methods either model the spread of two stocks or directly model the joint dynamic of stocks. Similar to Copula, an alternative method to model multivariate asset returns is by using the MBG Model discussed in Chapter 1. The MBG model is most suited to model multivariate financial assets return and has the advantage of capturing the negative skewness, extra kurtosis, the asymmetry in the upward and downward motions, and the tail dependence structure of assets returns nicely. Since one key component of pair trading strategies relies on how well the model tracks the stock’s comovement, this makes MBG an excellent model to build pair trading strategy. In this paper, we focus on distance, cointegration, copula, and MBG methods that we will discuss in the following section.

The contributions of this paper to the relevant literature are three-fold. First, we propose using a new method, the MBG method, to model the dependency structure of stock pairs and describe a practical approach for generating mispricing measures of stocks and

trading strategies. In the MBG method, we begin with fitting the two stocks in stock pairs to the BG marginal; then we model their joint distribution using the MBG, followed by trading signal generation calculated using the conditional probability distribution of stock returns. This novel method provides an alternative approach for constructing a strategy to trade correlated stocks. Second, we comprehensively study the four pair trading strategies—distance, cointegration, copula, and the MBG methods. The most recent study that compares distance, cointegration, and copula method is by [Rad et al. \(2016\)](#), which used the US equity data from 1962–2014. We extend this study in a similar setting by testing our strategies against a dataset consisting of all S&P500 stocks from 2016 to 2019. Third, we evaluate the performance of the four trading strategies with various risk-adjusted measures, including measures utilizing lower and higher partial moments. We also evaluate if the profit from pair trading is associated with various risk factors in the equity market by regressing the excess return against Fama–French–5 research factors ([Fama and French, 1993, 2015](#)), momentum, and mean-reversion factors.

To provide a comprehensive comparison between the four approaches to pair trading, we start by evaluating their performance by comparing their excess return. We report two kinds of returns—return on committed capital and return on fully-invested capital, with or without transaction cost—thus, there are a total of four returns. Despite the excess return, we also calculate various risk-adjusted measures, including measures that rely on lower/higher partial moments to evaluate their risk-adjusted return. The measures we report include Sharpe ratio, maximum drawdown, Omega ratio, Sortino ratio, Kappa–3 ratio, Gain–Loss ratio, and upside potential ratio. [Rad et al. \(2016\)](#) stated that the distance method and the cointegration methods suffer from a decline in trading op-

portunities. To retest this statement using recent data and gain a deeper understanding of our newly proposed MBG method, we also analyze the number of times each strategy has opened trade and their convergence rate. We also regress the daily excess return against common equity risk factors, including the Fama–French-5 research factors, to test if the profitability comes from different systematic risk factors. Moreover, the robustness of the excess return is estimated by the significance of the intercept of the regression.

The rest of this chapter is structured as follows. In Section 2, we review the literature on pair trading, cointegration, copula, and MBG process; Section 3 provides an in depth description of the methodology; Section 4 discusses the dataset and presents the results and key findings. Finally, the concluding remarks are in Section 5.

2.2 Literature Review

The first academic study that carried out a pair trading strategy was that by [Gatev et al. \(2006\)](#). In this study, the authors tested the most common and most straightforward trading rule—named distance method—against the Center for Research in Security Prices (CRSP) stock data from 1962 to 2002. Using the distance method, they achieved a monthly excess return of 1.4% for the top-20 pairs’ portfolios on average. By restricting trading pairs from the same industrial sector, they recorded a monthly excess return of 1.1%, 0.8%, 0.6%, and 0.6% from utilities, financial, transportation, and industrial sectors, respectively. [Do and Faff \(2010, 2012\)](#) extended this study using data from 1962 to 2010 and concluded that this trading strategy performed particularly strongly during market downturns. However, due to the increased divergence rates and market efficiency,

the monthly excess return for the top 20 pairs within the same industry group decreased by 76 basis points, from 102 basis points in 1962–1988 to 26 basis points in 2003–2009. Furthermore, with regard to dynamic transaction cost, this strategy became largely unprofitable after 2002. Other researchers also explored distance method trading strategies in different markets and asset classes: [Andrade et al. \(2005\)](#) tested it on data from the Taiwan Stock Exchange, [Perlin \(2009\)](#) tested it on data from the Brazilian financial market, [Bowen and Hutchinson \(2016\)](#) tested this method in the UK equity market, while [Jacobs and Weber \(2015\)](#) conducted a cross country study on 24 international stock markets and concluded that the excess return from this trading strategy is a persistent phenomenon that varies significantly over time. Their findings indicate that the source of profitability associated with this method comes from investors' attention and reaction to firm-specific news shocks.

Another classical method for constructing pair trading is using cointegration. Based on the error correction model representation of cointegrated time series presented by [Engle and Granger \(1987\)](#), [Vidyamurthy \(2004\)](#) presented a detailed framework for pair trading using cointegration, including equity pair selection, signal estimation, and trade design. [Lin et al. \(2006\)](#) proposed a five-step trading strategy called cointegration coefficient weighting strategy (CCW), which uses the cointegration coefficient to decide the proportion of long–short position to ensure that each trade delivers a minimum nominal profit per trade. This paper uses both simulated and actual data from two Australian bank shares to test the profitability of this strategy. [Caldeira and Moura \(2013\)](#) used the cointegration method to estimate the profitability of this strategy on the Sao Paulo stock exchange. Using data from January 2005 to October 2012, their research exhibited

an annualized excess return of 16.38%, Sharpe ratio of 1.34, and the excess return was uncorrelated with the market, indicating that this strategy is market neutral. [Huck and Afawubo \(2015\)](#) used the distance and cointegration methods to test on S&P 500 stocks from August 2000 to September 2011. After controlling for risk factors and transaction cost, they concluded that the cointegration-based method demonstrates high and consistent positive alpha, with excess return ranging from 1.38% to 5% per month. However, after adjusting for transaction cost, the distance method failed to generate significant and stable excess return. [Miao \(2014\)](#) extended this method to high-frequency trading and developed a dynamic pair trading system based on the cointegration approach to test on NASDAQ and NYSE energy sector stocks from 2012 to 2013. During the 12-month out-of-sample test period, this trading system recorded a monthly return of 3.82%, a monthly Sharpe ratio of 2.73, and revealed a low level of correlation with the S&P 500 index.

Copulas (which have the flexibility to select marginal distributions and model joint distribution separate from marginal) have been frequently used to model the high dimensional dependency structure in the financial industry, especially in risk management. [Xie and Wu \(2013\)](#) proposed the use of copula for pair trading using the mispricing index as a trading signal and further proved this index to be a valid measure for the degree of mispricing in stocks. They also proved that the distance and cointegration methods could be generalized as special cases of the copula method under certain dependency structures. [Liew and Wu \(2013\)](#) utilized the five most commonly used copula as strategies candidate models to implement the copula method and determine the best-fitting one with information criteria. They compared the copula method against the distance and cointegration methods on three stock pairs from December 2009 to November 2012 and concluded that

the copula method is a potentially powerful alternative to the other two traditional pairs trading techniques. [Xie et al. \(2016\)](#) implemented a similar trading strategy and tested this method on a broader set of 89 US equities in the utility sectors from 2003 to 2012. The first academic work that performed a comprehensive evaluation of the copula method was by [Rad et al. \(2016\)](#). They tested the copula method, together with the distance and cointegration methods, on data consisting of all tradable stocks in the US market from 1962 to 2014 and concluded that compared to the other two methods, although the copula-based trading strategy has relatively lower return volatility and more trading opportunities, it fails to deliver robust excess returns after transaction cost is applied. The copula method for pair trading could be further divided into two subteams—return-based and level-based copula methods. Literature that implemented the return-based copula method includes [Ferreira \(2008\)](#), [Liew and Wu \(2013\)](#), and [Stander et al. \(2013\)](#). This subcategory directly uses the conditional marginal distribution function (calculated as the copula function's first partial derivative) of stock returns as a trading signal, i.e., enter a position when conditional probability is larger than 0.95 or smaller than 0.05. The level-based copula method was developed by [Xie and Wu \(2013\)](#), [Xie et al. \(2016\)](#), and [Rad et al. \(2016\)](#). In this approach, overall daily mispricing is calculated as the conditional marginal distribution minus 0.5. The cumulative mispricing index is defined by the summation of overall daily mispricing over historical periods. In contrast to the return-based approach, the level-based approach relies on cumulative mispricing as a trading signal instead of daily mispricing.

The MBG method is similar to the copula method, which directly models the co-movement of stock pairs instead of modeling the spread, such as in the distance and

cointegration methods. MBG (Madan, 2020) is an extension of MVG (Buchmann et al., 2019) to consist of predetermined BG marginals. The advantage of using MBG is that it could model the asymmetric in the upward and downward motions and the tail dependence structure of asset returns. In the MBG method, we use the cumulative mispricing index as a trading signal. We extend the mispricing index defined in the level-based copula method by introducing a lookback period. This lookback period is the hyperparameter that specifies how far back in the past mispricing information one wants to consider. In our implementation, we use the MBG model described in Chapter 1 and the CGMM with the reparameterization of the correlation matrix as the calibration procedure for fitting the model.

2.3 Research Method

For all four strategies, we use a consecutive period of 12 calendar months as the formation period and the following two calendar months as the testing period. Our dataset spans from January 2016 to December 2019; therefore, we end up with 17 overlapping formation periods and 17 nonoverlapping testing periods. Models and model parameters are calibrated using the data from the formation period, and performance is estimated solely using the data from the testing period. To avoid data snooping bias, candidate pairs are selected using the data from the formation period and nominated for trading in the testing period. We impose a 5% stop-loss threshold and a 10% stop-gain threshold for each trade to simulate the actual action of practitioners.

2.3.1 Distance method

Our implementation of the distance method is similar to that by [Gatev et al. \(2006\)](#).

At the beginning of the 12-month formation period, we first rescale the time series of adjusted prices for each stock so that the adjusted price is set to \$1 at the beginning of each formation period.

Let P_{it} be the adjusted stock price of the i^{th} stock at time t , where t ranges from 0 to T , 0 and T representing the beginning and end of the formation period, respectively. The rescaled price series, P^s , P_{it}/P_{i0} . After acquiring the rescaled adjusted price series, for each possible stock pair, we calculated their sum of squared differences (SSD), which we define as:

$$SSD_{i,j} := \sum_{t=0}^T (P_{it}^s - P_{jt}^s)^2 \quad (2.1)$$

where t ranges from 0 to T , i, j represent any two valid stocks, and P^s is the rescaled adjusted price of the i^{th} stock at the time t .

The SSD is a measure of the comovement of stocks. The lower the SSD, the more likely the two stocks will move together. For each trading period, we ranked the SSD from low to high for all stock pairs and selected the lowest 20 pairs to form the candidate pairs. We further define $Spread_{DM}$:

$$Spread_{DM_t} = P_{A_t}^s - P_{B_t}^s \quad (2.2)$$

as the difference of scaled price series for stock A, P_A^s , and stock B, P_B^s . Finally, the only parameter that we need to share with the testing period is σ_{DM} , which is calcu-

lated as the standard deviation of the $Spread_{DM}$ from the training period. We define $Spread_{standardized}^{DM}$ as $Spread_{DM}$ divided by σ_{DM} , i.e.,

$$Spread_{standardized}^{DM} := \frac{Spread_{DM}}{\sigma_{DM}}. \quad (2.3)$$

At the beginning of the testing period, prices for each stock are once again scaled to be \$1 and the standardized spread for each pair is also calculated and monitored. It is worth noting that using the σ_{DM} , we estimated from the formation period to standardize the testing period's $Spread_{DM}$. We set two threshold values, α_{DM}^{high} and α_{DM}^{low} and set them arbitrary to 2 and -2, respectively, as the signals for entering a position. If the standardized spread is more extreme than the two thresholds, we simultaneously open long and short positions, depending on the direction of the divergence. If the standardized spread is higher than 2, this suggests that stock A is relatively more expensive than stock B (since the spread is defined as the standardized price of stock A minus the standardized price of stock B). Therefore, we open a long position of \$1 worth of stock B and open a short position of \$1 worth of stock A at the same time. If the standardized spread is smaller than -2, we open two positions, but this time, we short \$1 worth of stock B and long \$1 worth of stock A. Trading positions are closed when the standardized spread reverts (cross zero), and we monitor the same pair for another potential trading opportunity.

2.3.2 Cointegration method

2.3.2.1 Framework

Let X_t be a nonstationary time series; X_t is called an $I(1)$ series if its first differences form a stationary time series, an $I(0)$ series. Assuming the price series of adjusted stock prices are the $I(1)$ series, we can define cointegration based on $I(1)$ series. Consider X_t and Y_t to be the two time series of adjusted stock price and assume they are both $I(1)$ series. If there exists a nonzero real number β such that:

$$Y_t - \beta X_t = u_t \quad (2.4)$$

and if u_t is an $I(0)$ series, then the two time series, X_t and Y_t are cointegrated time series (Lin et al., 2006).

Another equivalent representation of the cointegration relationship can be shown via the error correction model (Vidyamurthy, 2004) using the Granger representation theorem (Engle and Granger, 1987). The idea behind error correction is that the cointegrated systems have an equilibrium value equal to their long-run mean, which can be formulated as the a linear combination of two time series. The two time series will restore themselves to their long-run equilibrium of zero if they deviate from their long-run means. Let ϵ_{yt} and ϵ_{xt} be two white noise processes corresponding to the time series, Y_t and X_t ; the error correction representation is:

$$y_t - y_{t-1} = \alpha_y(y_{t-1} - \beta * x_{t-1}) + \epsilon_{yt}; \quad (2.5)$$

$$x_t - x_{t-1} = \alpha_x(y_{t-1} - \beta * x_{t-1}) + \epsilon_{xt}. \quad (2.6)$$

For the first equation, the left-hand side of the equation, $y_t - y_{t-1}$, represents the innovation of the times series, Y_t , for each time step, while the right-hand side can be separated into two parts—the error correction part, $\alpha_y(y_{t-1} - \beta x_{t-1})$ and the white noise part, ϵ_{yt} . In the error correction part, β is the coefficient of cointegration and $(y_{t-1} - \beta x_{t-1})$ represents the magnitude as well as the direction of deviation from the long-term mean 0. The error correction rate, α_y , indicates the speed of correction. The higher the α , the faster the time series adjust itself back to equilibrium. Under this framework, the two time series evolve in time, deviate from their equilibrium caused by the error term, and restore themselves to an equilibrium value. In accordance with [Rad et al. \(2016\)](#), we define the spread series as the difference between y_t and βx_t , such as:

$$Spread_{CO} = Y_t - \beta X_t. \quad (2.7)$$

We further define the normalized spread of cointegrated time series, $Spread_{normalized}^{CO}$, as:

$$Spread_{normalized}^{CO} = \frac{(Spread_{CO} - \mu_{CO})}{\sigma_{CO}}, \quad (2.8)$$

where μ_{CO} is the mean of the spread and σ_{CO} is the standard deviation of the spread.

[Rad et al. \(2016\)](#) defined the payoff of trade as the change in spread series. Suppose we purchase one share of stock, Y, and short β share of stock, X, at a time, $t - 1$. The

payoff of this trade at the next time step, t , is:

$$(Y_t - Y_{t-1}) - \beta(X_t - X_{t-1}). \quad (2.9)$$

Rearranging the terms in the above equation yields the following:

$$(Y_t - \beta X_t) - (Y_{t-1} - \beta X_{t-1}) = Spread_{CO_t} - Spread_{CO_{t-1}} \quad (2.10)$$

Therefore, the profit and loss of simultaneously longing one shares of stock, Y , and shorting β share of stock, X , in the time interval $[t - 1, t]$ is the change in spread series. By definition, the spread series is stationary; therefore, we can use these two facts to construct a trading strategy with a trading rule to open positions when spread series deviates from its long-term equilibrium and close positions as the spread series reverts to its long-term equilibrium.

2.3.2.2 Trading strategy

We adopt the two-step procedure in the pair selection process for cointegration pair trading strategy documented by [Rad et al. \(2016\)](#). The first step is the same as the procedure we use in the distance method. For each of the 12 months in the formation period, we calculate SSD using Equation (2.1) for all possible pairs using the adjusted price series from the formation period. We then rank the pairs in ascending order based on their SSD values. In the second step, we test for cointegration starting from the pair with the least SSD. If such a pair is cointegrated, we then estimate the corresponding cointegration

coefficient and add it to the list of candidate pairs. We continue this process until we have 20 pairs, which we monitor for trading opportunities during the testing period. We use the Engle–Granger two-step cointegration test (Engle and Granger, 1987) to test the cointegration for each pair and say the pair is cointegrated if the p -value from the test result is < 0.05 . We then use ordinary least squares (OLS) to get the corresponding cointegration coefficient β .

For each candidate pair we select during the formation period, we estimate the following three parameters and reuse them in the testing period: the cointegration coefficient β , the mean of the spread series μ_{CO} , and the standard deviation of the spread series σ_{CO} . Note that the μ_{CO} and σ_{CO} are estimated using the sample mean and sample standard deviation of the spread series we calculated from the adjusted price series in the formation period. At the beginning of the testing period, we reconstruct the normalized spread $Spread_{normalized}^{CO}$ using σ_{CO} , μ_{CO} and the adjusted price series from the data for each candidate pair during the testing period. We arbitrarily set the two values α_{high}^{CO} and α_{low}^{CO} to 2 and -2 as the thresholds to establish positions. For instance, for any date in the testing period, if the normalized spread drops below -2 for a specific pair that we do not have positions in, we long \$1 worth of stock, Y, and short β worth of stock, X. Since the spread is defined as $Y_t - \beta X_t$, a decrease in a spread indicates that Y_t becomes relatively more expensive than X_t , or X_t becomes relatively cheaper than Y_t . This CCW strategy is not dollar-neutral; therefore, it is not a self-financing portfolio. Conversely, if the normalized spread increase above 2, we long \$1 worth of stock, X, and short $\frac{1}{\beta}$ worth of stock, Y. We simultaneously close out our positions when the normalized spread crosses 0, indicating that it has reverted into its long-term mean. Similar to the distance method, the pair is

then monitored for other trading opportunities on the next business date.

In Figure 2.1, we plot the prices and spread of two stock pairs. The subplot on the upper-left contains the prices for the two stocks CMS and FE. This stock pair is cointegrated with the coefficient $\beta = 0.74$, and its spread μ is displayed in the lower-left subplot. The subplot on the upper-right contains the prices for the two stocks GS and MKTX that are not cointegrated, and its spread μ is displayed in the lower-right subplot.

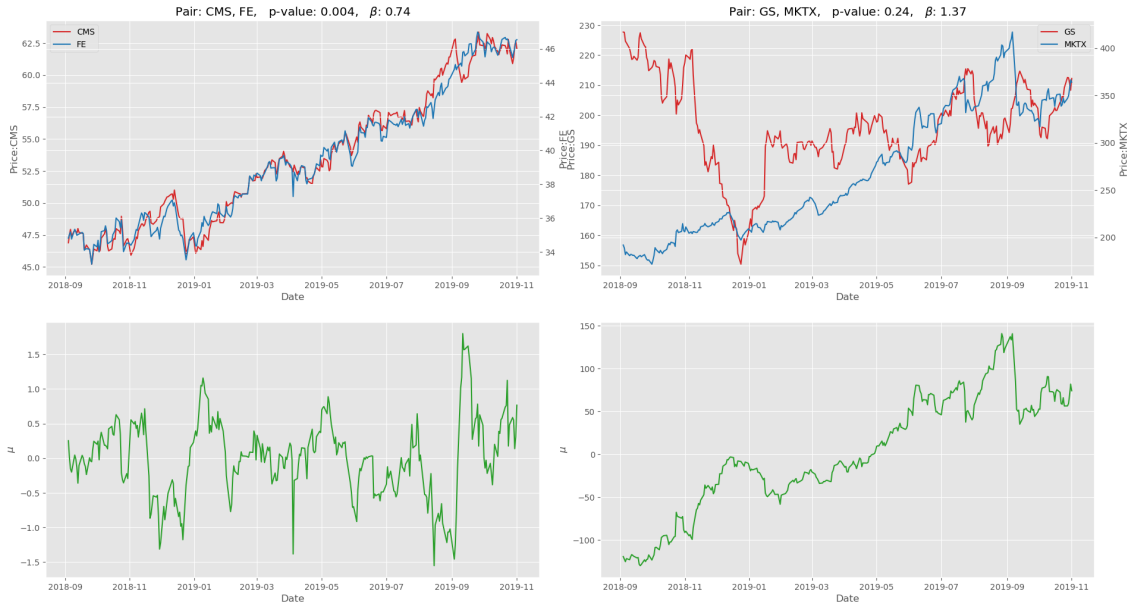


Figure 2.1: Cointegrated and non-cointegrated stock pairs

2.3.3 Copula method

2.3.3.1 Framework

It is clear from Definition 1.2.9 that a copula is nothing but a multivariate distribution with support in $[0, 1]^n$ and with uniform marginals. The fact that such copulas

can be very useful for representing multivariate distributions with arbitrary marginals is seen from Sklar's theorem (1.2.2), which provides both a parameterization of multivariate distributions and a construction scheme for copulas.

Recall Section 1.2.4 for the detailed introduction on copula. From Equation (1.8), we can see that a multivariate distribution could be decomposed into marginal probability density functions and copula density functions. Therefore, copula allows for high flexibility in the modeling of multivariate distribution, thereby enabling the flexible modeling of each component's marginal distribution independently. Moreover, since the choice of copula is independent of the marginal distributions, one could select different copulas to model different dependence structures. Among the different copulas, one of the most commonly used copula in the financial industry is the Gaussian copula (1.2.10) in which the following proposition leads to the trading signal in the copula method from B Sabino da Silva et al. (2017):

Proposition 2.3.1. Let U_1 and U_2 be two random variables with distribution $U(0, 1)$. Then,

$$P(U_1 \leq u_1 \mid U_2 = u_2) = \frac{\partial C(u_1, u_2)}{\partial u_2} = P(X_1 \leq x_1 \mid X_2 = x_2);$$

$$P(U_2 \leq u_2 \mid U_1 = u_1) = \frac{\partial C(u_1, u_2)}{\partial u_1} = P(X_2 \leq x_2 \mid X_1 = x_1),$$

where

$$\frac{\partial C(u_1, u_2)}{\partial u_2} = \lim_{h \rightarrow 0} P(U_1 \leq u_1 \mid u_2 \leq U_2 \leq u_2 + h) \quad (2.11)$$

$$\frac{\partial C(u_1, u_2)}{\partial u_1} = \lim_{h \rightarrow 0} P(U_2 \leq u_2 \mid u_1 \leq U_1 \leq u_1 + h). \quad (2.12)$$

For the Gaussian copula, we have (Liew and Wu, 2013):

$$P(U \leq u \mid V = v) = \frac{\partial C}{\partial v} = \Phi \left(\frac{\Phi^{-1}(u) - \rho \Phi^{-1}(v)}{\sqrt{1 - \rho^2}} \right); \quad (2.13)$$

$$P(V \leq v \mid U = u) = \frac{\partial C}{\partial u} = \Phi \left(\frac{\Phi^{-1}(v) - \rho \Phi^{-1}(u)}{\sqrt{1 - \rho^2}} \right), \quad (2.14)$$

where $\rho \in (-1, 1)$ is the correlation coefficient and Φ is the standard normal distribution function.

We could get a conditional distribution function from the partial derivative of the copula function and Xie et al. (2016) defined a measure, called the mispricing index, to describe the degree of mispricing between two financial assets.

Let P_t^x, P_t^y be the daily closing prices for stocks, X and Y , and let R_t^x, R_t^y be the daily returns for stocks, X and Y , on day t . For comparability with Gatev et al. (2006), returns and prices are related using:

$$R_t^X = (P_t^X - P_{t-1}^X) / P_{t-1}^X,$$

$$R_t^Y = (P_t^Y - P_{t-1}^Y) / P_{t-1}^Y.$$

Definition 2.3.1 (Mispricing index). Let R_t^X and R_t^Y represent the random daily returns of stocks, X and Y , at a time, t , and r_t^X and r_t^Y represent the realizations of those returns at a time t . Then define

$$MI_{X|Y}^t = \frac{\partial C(u_1, u_2)}{\partial u_2} = P(R_t^X < r_t^X \mid R_t^Y = r_t^Y) \quad (2.15)$$

and

$$MI_{Y|X}^t = \frac{\partial C(u_1, u_2)}{\partial u_1} = P(R_t^Y < r_t^Y \mid R_t^X = r_t^X) \quad (2.16)$$

where $u_1 = F_X(r_t^X)$ and $u_2 = F_Y(r_t^Y)$.

Therefore, the mispricing index $MI_{X|Y}^t$, indicates whether the return on a stock X , is high or low at a time t , conditional on the return of Y at a time t . The value of $MI_{X|Y}^t$ is a conditional probability and should range from 0 to 1; a high value is interpreted as, at time t , conditional on stock Y 's return, the probability that stock X 's historical return (R_t^X) is smaller than stock X 's current realized return (r_t^X) is high. This indicates that stock X is overvalued at time t compare with stock Y and the chances for the stock price of X to fall below its current realization is high. [Xie and Wu \(2013\)](#) proved that under fair pricing, $MI_{X|Y}^t$ is 0.5. Therefore, a value above 0.5 means stock X is overvalued at a time t , while a value below 0.5 means stock X is undervalued at a time t .

Note that $MI_{X|Y}^t$ and $MI_{Y|X}^t$ only measure the extent of relative mispricing for a single day t . To get a measurement of mispricing over a period of time, we first define $m_{1,t}$ and $m_{2,t}$ as the overall mispricing indexes of stock X and stock Y at time t , where:

$$\begin{aligned} m_{1,t} &= MI_{X|Y}^t - 0.5, \\ m_{2,t} &= MI_{Y|X}^t - 0.5. \end{aligned} \quad (2.17)$$

We then define $M_{1,t}$ and $M_{2,t}$ as the two cumulative mispricing indexes for stocks X and Y by:

$$\begin{aligned} M_{1,t} &= M_{1,t-1} + m_{1,t} \\ M_{2,t} &= M_{2,t-1} + m_{2,t} \end{aligned} \quad (2.18)$$

and $M_{1,0} = 0$ and $M_{2,0} = 0$. By construction, the two cumulative mispricing indexes are set to zero at the beginning of the trading period, and the level of mispricing is accumulated by summing the overall mispricing index m as time passes.

2.3.3.2 Trading strategy

There are two popular approaches for pair selection for the copula method. The first is the pure copula approach ([Krauss, 2017](#)), which tests all possible pairs and selects the top-performing pairs that pass the predefined correlation threshold in the formation period. The second is the SSD approach, which is the same as we previously described for the pair selection for the distance method. At first glance, the pure copula approach seems to be a superior methodology because this integrated approach uses both copulas for selecting pairs and trading. However, the most significant drawback is computational inefficiency. Since it needs to test for all possible pairs (in our case, there are $500 \times [500 - 1]/2$ pairs) in the formation period, it is almost impossible in practice. Therefore, we use the SSD approach to select the top 20 pairs in each of the formation periods.

The advantage of the copula method is that we have the flexibility to model the marginal distributions independent of the joint distribution. Considering that financial asset return is heavy-tailed and has excess kurtosis compared to other distributions, the BG distribution is one of the superior choices for modeling stock returns ([Madan and Wang, 2017](#)). Hence, in our implementation of the copula-based trading strategy, we use the BG distribution as the marginal distribution for each stock pair. In the formation period, for each candidate pair, we first fit the daily returns of each stock to the BG

distribution and estimate the parameters, b_p , c_p , b_n , and c_n , for each stock. In the second step, we model the joint distribution using the Gaussian copula together with the estimated parameters of the marginal model. The estimated parameters of Gaussian copula and BG marginals from the formation period will be stored and reused in the testing period.

For each day during the testing period, using the actual daily returns, r_t^x and r_t^y , we calculate the mispricing indexes using Equations (2.15) and (2.16). Inserting the results from Equations (2.15 and 2.16) into Equations (2.17) and (2.18), respectively, we further calculate the daily overall mispricing indexes, $m_{1,t}$ and $m_{2,t}$, and the daily cumulative mispricing indexes, $M_{1,t}$ and $M_{2,t}$. We define two thresholds, α_{CP}^{high} and α_{CP}^{low} , and set them to 0.5 and -0.5 , respectively, in accordance with Rad et al. (2016). For the pair we do not have positions in, if we observe that $M_{1,t} > \alpha_{CP}^{high}$ and $M_{2,t} < \alpha_{CP}^{low}$ on the same day—which suggests that stock X is overvalued and stock Y is undervalued—we construct a self-financing portfolio consisting of a \$1-long position in stock Y and a \$1-short position in stock X. Conversely, if we observe that $M_{2,t} > \alpha_{CP}^{high}$ and $M_{1,t} < \alpha_{CP}^{low}$ on the same day—which suggests that stock Y is overvalued and stock X is undervalued—we construct a self-financing portfolio consisting of a \$1-long position in stock X and a \$1-short position in stock Y. These positions will be unwound after both cumulative mispriced indexes return to zero. After closing out the positions, the same pair is then monitored for other trading opportunities starting the next trading day.

2.3.4 MBG method

2.3.4.1 Framework

Recall Section 1.3.1 for the framework of MBG. We use the CGMM method discussed in Section 1.3.3 to estimate the MBG parameters. To simulate data under the MBG model, we use the sequential sampling technique. The simulation scheme is presented in Section 1.3.2. Using this simulation method, we can calculate the conditional probability that will be needed in the next sub-section.

Let X_1, X_2 be the two components of a two-dimensional MBG process. To calculate the conditional probability $P(X_1 < x_1 \mid X_2 = x_2)$ for specific x_1 and x_2 , we first sample n pairs of data using the above algorithm; then we count the number of pairs with the second component, X_2 larger than $x_2 - \epsilon$ and smaller than $x_2 + \epsilon$. If we have m number of such pairs, we further count the number of pairs with the first component, X_1 less than x_1 , and denote that as k . We then have the conditional probability as the ratio of k and m as:

$$P(X_1 < x_1 \mid X_2 = x_2) = \frac{k}{m} \quad (2.19)$$

Unlike the mispricing indexes (2.15) and (2.16) in the copula method, which is calculated as the partial derivative of the copula function, we define the mispricing index for the MBG method, $MI_{X|Y}^t$ and $MI_{Y|X}^t$ as:

$$\begin{aligned} MI_{X|Y}^t &= P(X_1 < x_1 \mid X_2 = x_2), \\ MI_{Y|X}^t &= P(X_2 < x_2 \mid X_1 = x_1), \end{aligned} \quad (2.20)$$

where the conditional probability is calculated by using the above algorithm.

Similar to the copula method, we further define the daily overall mispricing indexes, $m_{1,t}$ and $m_{2,t}$, to describe the overall mispricing of two stocks for any given day. We then define the cumulative mispricing index with lookback period as:

$$M_{1,t} = \sum_{i=\max(0,t-L+1)}^t m_{1,i} \quad (2.21)$$

$$M_{2,t} = \sum_{i=\max(0,t-L+1)}^t m_{2,i},$$

where L is the lookback period in days and $M_{1,0} = 0$ and $M_{2,0} = 0$.

By construction, at the beginning of the trading period, the two cumulative mispricing indexes are set to zero and the level of mispricing accumulates as time passes by adding the overall mispricing index m to itself.

2.3.4.2 Trading strategy

The candidate pair selection process for the MBG method is the same as the process for the distance method. We first calculate SSD for all possible pairs and select the 20 pairs with the smallest SSD as candidate pairs. In the formation period, for each stock pair, we first use the Fourier inversion technique to calibrate the two BG marginals, then model their joint dynamic by fitting to the MBG model using the CGMM method. For each stock pair, we store their corresponding parameters θ_{MBG} for the calculation of

trading signal in the testing period.

$$\theta_{MBG} = (b_p^x, b_n^x, c_p^x, c_n^x, b_p^y, b_n^y, c_p^y, c_n^y, \nu, \rho),$$

where x and y are the two stocks in the same pair.

For each trading day in the testing period, for each candidate pair, we first calculate its mispricing indexes, $MI_{X|Y}^t$ and $MI_{Y|X}^t$, from the actual daily returns, r_t^x and r_t^y , using the Monte Carlo method. We further calculate the overall mispricing indexes, m_t^1 and m_t^2 , and the cumulative mispricing index with lookback periods, $M_{1,t}$ and $M_{2,t}$. In our implementation, the lookback period, L , is set as 20 trading days because we believe the information on mispricing 1 month ago should have no impact on the cumulative mispricing for today. We define two threshold variables, α_{MBG}^{high} and α_{MBG}^{low} , and set them arbitrarily to 0.5 and -0.5 , respectively, if we observe that $M_{1,t} > \alpha_{MBG}^{high}$ and $M_{2,t} < \alpha_{MBG}^{low}$ for the pair for which we have no opening trade. We construct a self-financing portfolio of \$1-long position in stock Y and \$1-short position in stock X since the cumulative mispricing indexes indicated that stock X is overvalued relative to stock Y . In contrast, if we have that $M_{2,t} > \alpha_{MBG}^{high}$ and $M_{1,t} < \alpha_{MBG}^{low}$ —which suggests that stock X is undervalued relative to stock Y —we simultaneously long \$1 of stock X and short \$1 of stock Y . We will unwind these two positions when both cumulative mispricing indices revert to zero. Upon closing this trade, we monitor the same pair for other trading opportunities starting the next trading day.

2.4 Data and Empirical Result

2.4.1 Data

Our dataset consists of daily closing prices of S&P500 in the CRSP from January 2016 to December 2019. This dataset contains a total of 1188 trading days and 578,556 daily prices. Since the raw data in the CRSP is not adjusted for dividend, split, and corporate action, we perform the adjustments ourselves using the cumulative adjustment factor provided in the CRSP to get the adjusted prices. We use the highest level of the Global Industry Classification Standard code maintained by MSCI.Inc to bucket stocks into 11 distinct sectors. For the daily factors data, we pull that data from Professor Kenneth French's website ([French, 2022](#)).

2.4.2 Transaction cost

Transaction cost plays a vital role in estimating the profitability of pair trading strategies. Commissions have decreased significantly over the last 50 years. Average transaction costs for US Equity decreased from 70 basis points in 1963 to only 9 basis points in 2009 ([Do and Faff, 2012](#)). Given that our dataset goes beyond 2009 and the transaction cost has even decreased further, we use a one-way transaction cost of 5 basis points in this paper. Since we need to trade two stocks simultaneously for both opening and closing of trade, the transaction cost for one trade is 20 basis points in total.

2.4.3 Return calculation methodology

According to [Gatev et al. \(2006\)](#), there are two types of returns that can be calculated: return on committed capital (2.23) and the return on fully-invested capital (2.22). Return on committed capital is calculated as the summation of the returns on each pair divided by the number of stock pairs selected, even if some of the pairs do not trade in the testing period. Conversely, returns on fully-invested capital is calculated as the summation of the returns on each pair divided by the number of stock pairs that actually trade in the testing period. Compared to returns on fully-invested capital, return on committed capital is a more conservative measure that takes into account the opportunity cost of the capital allocated to trading. In our analysis, we adopt both methods for our return calculation. Furthermore, since our testing period is set as 2 months, we use Equation (2.24) to adjust the 2-month return, R_{2m} , to the annual return, R_{1y} . Excess return is calculated by subtracting the annualized risk-free rate implied by the US 3-month treasury bill from the annualized nominal return.

$$R_{2m}^{EC} = \frac{\sum_{i=1}^n r_i}{n} \quad (2.22)$$

$$R_{2m}^{CC} = \frac{\sum_{i=1}^n r_i}{NP}, \quad NP = 20 \quad (2.23)$$

$$R_{1y} = (1 + R_{2m})^6 - 1 \quad (2.24)$$

For the distance, copula, and MBG methods, we construct a \$1 long–short portfolio for each pair trade. For the \$1 long–short portfolio, we simultaneously open a long and

short position worth \$1. Since the \$1-long position can be fully financed by the money received from shorting the \$1 position, this portfolio is self-financing and, hence, does not require any capital for trading. However, this poses a problem in the return calculation. In this chapter, for the \$1 long–short portfolio, we assume the capital to be \$1, which is equal to the long leg of the trade. For the cointegration method, the long and short positions are not the same amount because the long position is \$1 and the short position varies by our estimated cointegration coefficient. However, since the long position is \$1 for all four methods, we also assume the capital used by the cointegration method is \$1, equal to the long position of the trade.

2.4.4 Descriptive statistics of portfolio excess return

To better understand the relative performance of these four trading strategies, we calculate the descriptive statistics for our nonoverlapping annualized 2-month testing period portfolio returns from 2016 to 2019. To estimate the robustness of the excess return, we also perform t-test to test if the excess return is significantly different from zero. Moreover, we test the normality of portfolio excess returns with Shapiro–Wilk test, which reported in Table 2.6.

Table 2.1 reports the annualized excess return for the four strategies, both before and after applying the transaction cost. Section 1 of the table reports the return on committed capital, while Section 2 shows the return on fully-invested capital. In general, the distance and cointegration methods have fewer trading opportunities than the other two methods (Table 2.4), which indicates that their return on fully-invested capital tends

Strategy	Mean	t-stat	p-val	Std. Dev.	Skewness	Kurtosis	VAR(5%)	Max	Min
Section 1: Return on Committed Capital									
<i>Panel A: After transaction cost</i>									
Distance	-0.012	-2.251	0.038	0.023	-0.499	-0.599	-0.052	0.020	-0.060
Cointegration	0.016	1.919	0.072	0.034	1.118	1.634	-0.018	0.109	-0.047
Copula	-0.010	-1.092	0.290	0.036	-0.276	-0.685	-0.062	0.055	-0.088
MBG	0.024	3.297	0.004	0.030	-0.263	-1.079	-0.022	0.070	-0.026
<i>Panel B: Before transaction cost</i>									
Distance	-0.007	-1.346	0.196	0.022	-0.347	-0.802	-0.042	0.030	-0.050
Cointegration	0.026	2.969	0.009	0.036	1.183	1.389	-0.009	0.122	-0.036
Copula	0.007	0.784	0.444	0.037	-0.159	-0.712	-0.044	0.076	-0.070
MBG	0.038	4.978	0.000	0.032	-0.467	-1.018	-0.014	0.089	-0.016
Section 2: Return on Fully Invested Capital									
<i>Panel A: After transaction cost</i>									
Distance	-0.004	-0.328	0.747	0.052	0.349	0.098	-0.068	0.120	-0.100
Cointegration	0.031	2.786	0.013	0.046	1.011	1.182	-0.016	0.142	-0.056
Copula	-0.008	-0.900	0.381	0.038	-0.217	-0.824	-0.063	0.058	-0.088
MBG	0.030	3.596	0.002	0.034	-0.176	-0.781	-0.026	0.092	-0.030
<i>Panel B: Before transaction cost</i>									
Distance	0.009	0.737	0.471	0.053	0.174	-0.153	-0.058	0.130	-0.090
Cointegration	0.046	3.965	0.001	0.048	1.062	1.072	-0.002	0.159	-0.042
Copula	0.009	0.951	0.355	0.039	-0.126	-0.849	-0.044	0.081	-0.070
MBG	0.047	5.594	0.000	0.035	-0.401	-0.829	-0.013	0.101	-0.014

Table 2.1: Mean return

to overestimate their return if the opportunity cost of the capital is considered. Before applying the transaction cost, all four strategies result in a positive excess return on fully-invested capital, while all other strategies except the distance method result in a positive excess return on committed capital. From the cumulative return in Figure 2.2 and Table 2.1, we can tell that the performance of the MBG method is similar to the cointegration method. In contrast, the performance of the distance method is more comparable to the copula method. The distance and the copula methods seem unprofitable, since their excess returns are negative in most scenarios. Even when excess returns are positive, the t-statistics are small in magnitude, indicating that the excess returns are not significantly different from zero. Excess returns for the cointegration and MBG methods remain positive and significant for all four scenarios. Their returns on fully-invested capital are

almost identical both before and after applying the transaction cost (differ by 10 basis points); however, for return on committed capital, MBG outperforms the cointegration method significantly. This outperformance comes from the fact that MBG has more trading opportunities in general, and hence, more pairs will be traded in the testing period (we will see that in Section 2.4.6). We also observe that all four strategies report similar standard deviation in return, and only the cointegration method systematically reports positive skewness and positive kurtosis. Of the four scenarios, we believe that excess return on committed capital after applying the transaction cost (2.1 section 1, panel A) is the most genuine measure of returns, since it takes into account both the cost of trading and the opportunity cost of unused capital. Based on these scenarios, we can say that the MBG method performs best, followed by the cointegration and copula methods, and that the distance method has the worst performance.

From the cumulative return plot, we can see that both MBG and cointegration methods maintain an upward trend. However, it is relatively hard to argue that the distance and the copula methods maintain positive returns.

2.4.5 Risk-adjusted performance

The Sharpe ratio, initially introduced by Sharpe (1966), is still one of the most famous risk-adjusted performance measures. The Sharpe ratio is calculated by dividing the portfolio excess returns by the volatility of returns, where the excess return is calculated by subtracting the risk-free rate from the expectation of the portfolio return. Other risk-adjusted measures have been proposed in recent years based on partial moments.

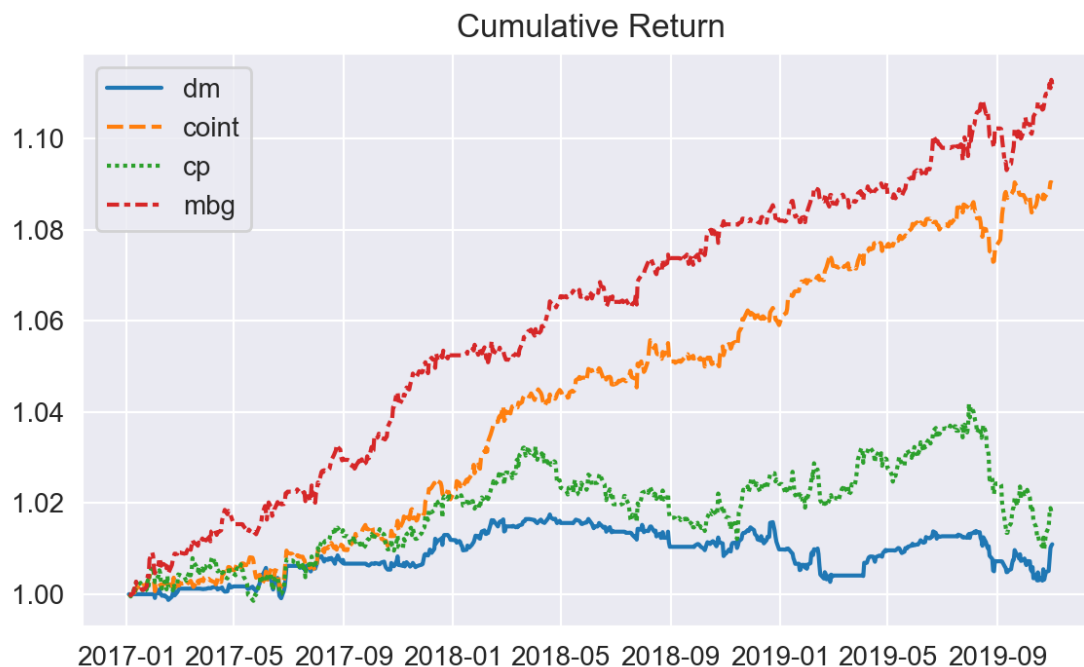


Figure 2.2: Cumulative return

Compared to the Sharpe ratio, lower partial moment measure only takes into account the negative deviation of returns below a certain threshold, whereas the Sharpe ratio takes into account both the positive and negative deviations. Omega ratio is defined as the ratio of returns above threshold versus returns below the threshold. Sortino ratio is defined as the ratio of the mean excess return to the negative standard deviation. Kappa ratio is a generalization of Omega and Sortino ratio introduced in [Kaplan and Knowles \(2004\)](#). The ratio we use here is Kappa-3 ratio, defined as the ratio of mean excess return to the third lower partial moment. Apart from measures that take lower partial moment into account, we also consider the gain–loss ratio and upside potential ratio that also take into account the upper partial moment. Gain–loss ratio, first presented by [Bernardo and Ledoit \(2000\)](#), is the ratio of the first upper partial moment of returns (upside potential) to the first lower partial moment (downside risk). The upside potential ratio, introduced by [Sortino et al.](#)

(1999), is the ratio of the first upper partial moment of returns to the square root of the second lower partial moment of returns (downside deviation). Additionally, we also consider the maximum drawdown, which is defined as the maximum observed loss from peak to trough of a portfolio in percentage points. It is a popular measure of downside risk in a portfolio.

Definition 2.4.1 (Lower Partial Moments). If τ is a chosen reference level, n is the degree of the moment, and X is a random variable with cumulative distribution $F_X(x)$, the lower partial moments (LPM) are given by

$$LPM_{n,\tau}(F_X) = E(\max(\tau - X, 0)^n) = \int_{-\infty}^{\tau} (\tau - x)^n dF_X(x) \quad (2.25)$$

In practice, LPM for m observations can be estimated by the following expression:

$$LPM_{n,\tau,i}^* = \frac{1}{m} \sum_{t=1}^m (\max(0, \tau - X_{i,t}))^n \quad (2.26)$$

LPM is a family of risk measures specified by τ and n . Often τ is set to the risk-free rate or simply to zero. By choosing the degree of the moment, an investor can specify the measure to suit their risk aversion. Intuitively, large values of n will penalize large deviations more than low values. Semivariance is a special case of LPM for which the degree of the moment is set to 2.

Similarly, upper partial moments can be defined as follows:

Definition 2.4.2 (Upper Partial Moments). If τ is a chosen reference level, n is the degree of the moment and X is a random variable with cumulative distribution $F_X(x)$, the upper

partial moments (UPM) are given by

$$UPM_{n,\tau}(F_X) = E(\max(X - \tau, 0)^n) = \int_{\tau}^{\infty} (x - \tau)^n dF_X(x) \quad (2.27)$$

In practice UPM for m observations can be estimated by the following expression:

$$UPM_{n,\tau,i}^* = \frac{1}{m} \sum_{t=1}^m (\max(0, X_{i,t} - \tau))^n. \quad (2.28)$$

Other measures mentioned above are defined as follows:

- Omega ratio

$$\mathbb{O}mR_y\{Y\} = \frac{E\{Y\} - y}{LPM_y^1\{Y\}}, \quad (2.29)$$

- Sortino ratio

$$\mathbb{S}oR_y\{Y\} = \frac{E\{Y\} - y}{\sqrt{LPM_y^2\{Y\}}}, \quad (2.30)$$

- Kappa-3 ratio

$$\mathbb{K}appa_{3,y}\{Y\} = \frac{E\{Y\} - y}{(LPM_y^3\{Y\})^{\frac{1}{3}}} \quad (2.31)$$

- Gain Loss Ratio

$$\frac{UPM_y^1\{Y\}}{LPM_y^1\{Y\}} \quad (2.32)$$

- Upside Potential Ratio

$$\frac{UPM_y^1\{Y\}}{\sqrt{LPM_y^2\{Y\}}} \quad (2.33)$$

We use the risk-free rate for the threshold value for calculating risk-adjusted mea-

Strategy	Sharp	Omega	Sortino	Kappa-3	GainLoss	UpsidePotential	Max Drawdown
<i>Panel A: After Transaction Cost</i>							
Distance	-1.359	-2.640	-1.439	-1.057	0.835	0.441	0.821
Cointegration	1.082	4.427	2.029	1.443	1.279	0.665	0.615
Copula	-0.378	-0.303	-0.328	-0.286	0.981	0.543	0.710
MBG	1.359	5.419	2.699	1.967	1.341	0.688	0.506
<i>Panel B: Before Transaction Cost</i>							
Distance	-0.740	-1.029	-0.591	-0.445	0.936	0.487	0.779
Cointegration	1.743	6.924	3.276	2.367	1.436	0.733	0.572
Copula	0.486	2.116	0.959	0.678	1.133	0.607	0.641
MBG	2.222	8.882	4.353	3.198	1.560	0.781	0.465

Table 2.2: Risk-adjusted return

asures related to lower or higher partial moments.

Except for the distance method, the three other methods record a positive Sharpe ratio before applying the transaction cost, while only MBG and cointegration methods record a positive Sharpe ratio after applying the transaction cost. The reason that copula's risk-adjusted return becomes negative is its insignificant positive mean return and highest number of trading opportunities. Therefore, it suffers a considerable decline in risk-adjusted return after applying the transaction cost. All strategies record a risk-adjusted performance decline after transaction cost is applied. The after transaction cost Sharpe ratio is approximately 60% of the previous cost value. Interestingly, the ranking of risk-adjusted return remains constant after cost and across all measures examined here. The MBG method records the best risk-adjusted performance, whereas the distance method records the worst one.

2.4.6 Properties of each pair trade

To further investigate the source of profit for the four pair trading strategies, we study the descriptive statistics and distribution of converged and unconverged trades. Ta-

ble 2.3 reports the statistics of each trade both before and after applying the transaction cost. Section 1 of the table reports the returns on all trades, while Section 2 only reports the trades that result in convergence. To get a more intuitive view of the distribution, we also plot the density of the returns on each trade (Figure 2.3) and each converged trade (Figure 2.4) after applying the transaction cost. Since we set up a 5% stop-loss threshold and a 10% stop-gain threshold, the lower bound of returns is -5% and the upper bound of returns is 10% before applying the transaction cost. Furthermore, since the transaction cost we apply is a flat 20 basis point cost, the before and after cost results in Table 2.3 are invariant for standard deviation, skewness, and kurtosis. At the same time, a constant shift for mean, maximum, and minimum is observed in the before and after costs results. In Section 1 of the table, the returns on each trade in all four methods are significant at a 5% level without transaction cost. However, after the transaction cost is applied, only the returns in the cointegration and the MBG methods remain significant. If we focus on only the trades that result in convergence, the returns in all four methods remain significant both before and after applying the trading cost. For converged trades, we observed that the distance and cointegration methods outperform the copula and MBG methods by around 80% in the mean return of each converged trade. However, based on Table 2.4, the MBG method has approximately seven times more converged trades than the distance method, and two times more converged trades than the cointegration method. The MBG method has the largest total return on converged trade, and this is one of the reasons the MBG method has the largest portfolio return.

Strategy	Mean	t-stat	p-val	Std. Dev.	Skewness	Kurtosis	VAR(5%)	Max	Min
Section 1: Return on Each Trade									
<i>Panel A: After transaction cost</i>									
Distance	0.001	0.798	0.426	0.022	-0.286	0.978	-0.040	0.059	-0.064
Cointegration	0.007	5.082	0.000	0.022	-0.013	0.496	-0.030	0.088	-0.060
Copula	0.001	0.797	0.426	0.021	-0.840	0.696	-0.044	0.058	-0.063
MBG	0.006	6.531	0.000	0.018	0.174	0.660	-0.021	0.064	-0.059
<i>Panel B: Before transaction cost</i>									
Distance	0.003	1.983	0.049	0.022	-0.286	0.978	-0.038	0.061	-0.062
Cointegration	0.009	6.536	0.000	0.022	-0.013	0.496	-0.028	0.090	-0.058
Copula	0.003	2.857	0.005	0.021	-0.840	0.696	-0.042	0.060	-0.061
MBG	0.008	8.779	0.000	0.018	0.174	0.660	-0.019	0.066	-0.057
Section 2: Return on Each Converged Trade									
<i>Panel A: After transaction cost</i>									
Distance	0.032	11.594	0.000	0.013	0.968	-0.350	0.019	0.059	0.018
Cointegration	0.034	22.258	0.000	0.012	1.519	4.762	0.017	0.088	0.015
Copula	0.017	25.287	0.000	0.009	1.254	2.406	0.006	0.058	0.002
MBG	0.017	14.581	0.000	0.015	0.212	0.635	-0.006	0.064	-0.034
<i>Panel B: Before transaction cost</i>									
Distance	0.034	12.309	0.000	0.013	0.968	-0.350	0.021	0.061	0.020
Cointegration	0.036	23.580	0.000	0.012	1.519	4.762	0.019	0.090	0.017
Copula	0.019	28.313	0.000	0.009	1.254	2.406	0.008	0.060	0.004
MBG	0.019	16.342	0.000	0.015	0.212	0.635	-0.004	0.066	-0.032

Table 2.3: Return of each trade

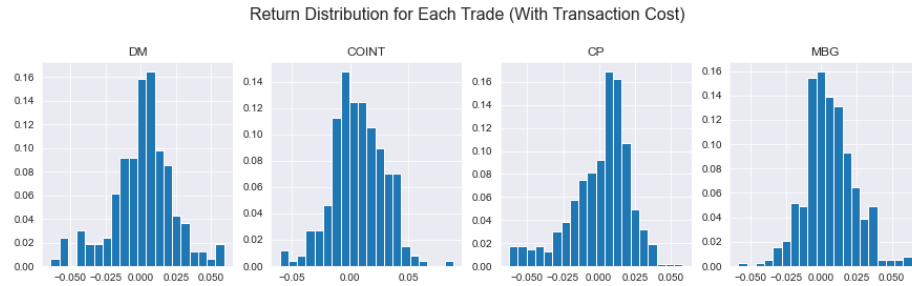


Figure 2.3: Return distribution for each trade (with transaction cost).

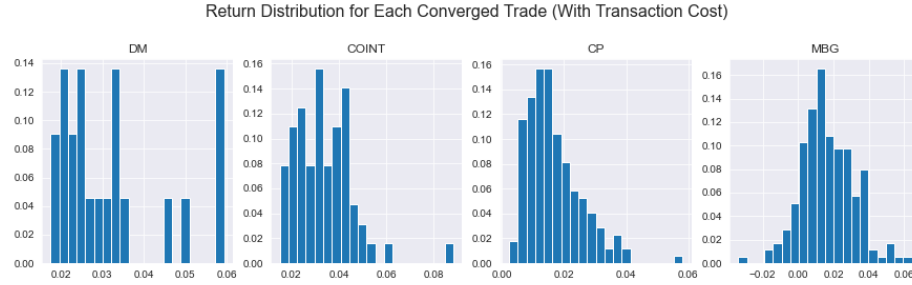


Figure 2.4: Return distribution for each converged trade (with transaction cost)

By the setup of the trading strategy, there are a total of four exit statuses—converge (COV), stop-loss (SL), stop-gain (SG), and unconverge (END) (force to exit the trade at the end of the testing period). Figure 2.5 displays the pie chart of exit status for each strategy. To further our study in trade convergence and trading opportunities, Table 2.4 reports the number of pairs traded for our nominated pairs and the number of times we open trade as well as the mean and median of days for trades to converge.

From our 20 nominated pairs for each testing period, the copula method trades 19.2 of them on average, while the distance method trades only 9. Together with the number of trades and fraction of trades that converged, it is not hard to see that the distance and cointegration methods have few trading opportunities, likely because these two methods are well studied and have been widely used by practitioners. New methods like the copula and the MBG methods have around 400 opportunities to open trade and 40% for trade convergence, while classical methods like distance and cointegration methods have only 200 opening trades and 20% convergence rate on average.

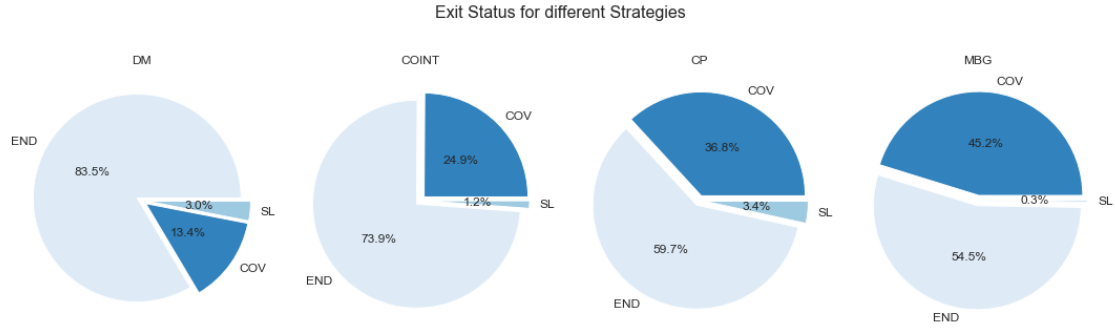


Figure 2.5: Exit status for different strategies

Strategy	# Traded Pairs	#Trades		Trade Type	Days.Cov. Mean	Days.Cov.Median
Distance	9/20	22	13.4%	Converge	15.2273	12.5
		142	86.6%	Other		
Cointegration	13.5/20	64	24.9%	Converge	16.0313	14.5
		193	75.1%	Other		
Copula	19.2/20	172	36.8%	Converge	12.7965	10.0
		295	63.2%	Other		
MBG	16.6/20	175	45.2%	Converge	15.9371	16.0
		212	54.8%	Other		

Table 2.4: Trade convergence analysis

2.4.7 Risk profile

To understand the source of the excess return for these pair trading strategies and determine whether the excess return is a compensation for various risk factors, we regress the daily excess return for our strategies against three popular sets of risk factors.

The first set of factors is the Fama–French 3 research factors (Fama and French, 1993), which contain the market return factor ($R_m - R_f$), size factor (small minus big[SMB]), and the value factor (high minus low[HML]). The market factor contains the excess return of a board market portfolio and is used to estimate the beta of portfolio return to market return. The size factor is the difference between the returns on a portfolio of stocks consisting of smaller companies and those consisting of big size companies. Similarly, the value factor is the returns on a portfolio of high book-to-market stocks minus the returns on a portfolio consisting of low book-to-market companies. Our second set of factors is the Fama–French 5 research factors (Fama and French, 2015), which add the profitability factor (robust minus weak [RMW]) and investment factor (conservative minus aggressive [CMA]) in addition to the Fama–French 3 research factor. The profitability factor is the difference of returns between robust operating profitability portfolios and weak operating profitability portfolios. The investment factor is the difference of returns between conservative investment portfolios and aggressive investment portfolios. Most trading strategies could be categorized as either mean revert or momentum strategies, and all the four strategies we study here place bets on either the mean revert of spread or mean revert of mispricing index. Therefore, we include momentum (Mom), short-term reversal (SRev), and long-term reversal (LRev) factors together with Fama–French 5 research factors as our third set of factors. The momentum factor is the return on a stock portfolio with a high 2–12 months prior period return minus the return on a stock portfolio with a low 2–12 months prior period return. Short-term reversal is defined as the return on a stock portfolio with a lower 1-month prior return minus the return on a stock portfolio with a higher 1-month prior return. Similar to the short-term reversal factor, instead of using the

return in the prior month, long-term reversal uses the prior 13–60 months’ prior returns to construct the stock portfolios.

Strategy	Annualized Excess Ret.	Annualized Alpha		Alpha	MKT	SMB	HML	RMW	CMA	Mom	ST_Rev	LT_Rev	R-Squared
Fama French 3 Factors													
Panel A: After Transaction Cost													
Distance	-0.0120	-0.0103	coef	-0.0041	-0.0019	-0.0017	0.0115						0.84%
			p-val	0.2196	0.6195	0.7952	0.0528*						
Cointegration	0.0160	0.0196	coef	0.0077	0.0044	0.0064	0.0276						3.27%
			p-val	0.024**	0.2780	0.3560	0***						
Copula	-0.0100	-0.0090	coef	-0.0036	0.0016	-0.0085	0.0088						0.19%
			p-val	0.4611	0.7914	0.3950	0.3253						
MBG	0.0240	0.0263	coef	0.0103	-0.0077	0.0010	-0.0167						1.01%
			p-val	0.0188**	0.1440	0.9085	0.0359**						
Panel B: Before Transaction Cost													
Distance	-0.0071	-0.0025	coef	-0.0010	-0.0017	-0.0010	0.0120						0.86%
			p-val	0.7751	0.6691	0.8807	0.0496**						
Cointegration	0.0310	0.0286	coef	0.0112	0.0048	0.0070	0.0284						3.27%
			p-val	0.001***	0.2378	0.3092	0***						
Copula	-0.0080	0.0086	coef	0.0034	0.0026	-0.0077	0.0083						0.19%
			p-val	0.4925	0.6649	0.4500	0.3631						
MBG	0.0300	0.0427	coef	0.0166	-0.0085	0.0021	-0.0182						1.01%
			p-val	0.0002***	0.1099	0.8187	0.0241**						
Fama French 5 Factors													
Panel C: After Transaction Cost													
Distance	-0.0120	-0.0093	coef	-0.0037	-0.0023	-0.0033	0.0098	-0.0217	0.0062				0.31%
			p-val	0.2648	0.5684	0.6212	0.1827	0.0277**	0.6188				
Cointegration	0.0160	0.0198	coef	0.0078	0.0026	0.0059	0.0321	-0.0074	-0.0135				3.52%
			p-val	0.0222**	0.5526	0.3966	0***	0.4710	0.2927				
Copula	-0.0100	-0.0088	coef	-0.0035	0.0001	-0.0091	0.0120	-0.0092	-0.0096				0.30%
			p-val	0.4780	0.9935	0.3647	0.2655	0.5359	0.6055				
MBG	0.0240	0.0271	coef	0.0106	-0.0089	-0.0010	-0.0166	-0.0265	0.0011				1.78%
			p-val	0.0148**	0.1105	0.9131	0.0827*	0.0451**	0.9447				
Panel D: Before Transaction Cost													
Distance	-0.0071	-0.0013	coef	-0.0005	-0.0021	-0.0027	0.0100	-0.0235	0.0065				1.88%
			p-val	0.8775	0.6102	0.6905	0.1827	0.0212**	0.6117				
Cointegration	0.0310	0.0289	coef	0.0113	0.0029	0.0065	0.0331	-0.0079	-0.0142				3.52%
			p-val	0.0009***	0.5087	0.3480	0***	0.4445	0.2701				
Copula	-0.0080	0.0091	coef	0.0036	0.0012	-0.0086	0.0111	-0.0114	-0.0079				0.30%
			p-val	0.4733	0.8564	0.4054	0.3136	0.4505	0.6730				
MBG	0.0300	0.0438	coef	0.0170	-0.0099	-0.0003	-0.0182	-0.0298	0.0012				1.78%
			p-val	0.0001***	0.0795*	0.9715	0.0597*	0.026**	0.9419				
Fama French 5 Factors + Momentum + Short Term Reversal + Long Term Reversal													
Panel E: After Transaction Cost													
Distance	-0.0120	-0.0095	coef	-0.0038	-0.0033	-0.0041	0.0091	-0.0212	0.0063	-0.0007	0.0059	0.0004	0.39%
			p-val	0.2560	0.4402	0.5578	0.2600	0.0486**	0.6621	0.9090	0.3716	0.9726	
Cointegration	0.0160	0.0198	coef	0.0078	0.0000	0.0009	0.0251	-0.0105	-0.0186	-0.0125	0.0109	0.0035	4.84%
			p-val	0.0227**	0.9945	0.8955	0.0016***	0.3510	0.2141	0.0512*	0.1194	0.7395	
Copula	-0.0100	-0.0095	coef	-0.0038	0.0007	-0.0072	0.0132	-0.0112	-0.0020	0.0010	0.0023	-0.0094	0.38%
			p-val	0.4394	0.9162	0.4935	0.2523	0.4873	0.9283	0.9144	0.8187	0.5385	
MBG	0.0240	0.0240	coef	0.0094	-0.0135	0.0006	-0.0122	-0.0202	0.0092	0.0146	0.0339	-0.0048	4.39%
			p-val	0.0295**	0.0203**	0.9488	0.2254	0.1553	0.6306	0.0761*	0.0001***	0.7225	
Panel F: Before Transaction Cost													
Distance	-0.0071	-0.0015	coef	-0.0006	-0.0033	-0.0041	0.0087	-0.0227	0.0057	-0.0015	0.0065	0.0015	2.10%
			p-val	0.8656	0.4476	0.5710	0.2893	0.0404**	0.7014	0.8203	0.3366	0.8854	
Cointegration	0.0310	0.0289	coef	0.0113	0.0001	0.0013	0.0259	-0.0109	-0.0201	-0.0131	0.0114	0.0042	4.84%
			p-val	0.0009***	0.9777	0.8569	0.0012***	0.3330	0.1812	0.0427**	0.1048	0.6876	
Copula	-0.0080	0.0081	coef	0.0032	0.0016	-0.0071	0.0113	-0.0141	0.0000	-0.0008	0.0039	-0.0101	0.38%
			p-val	0.5209	0.8155	0.5069	0.3354	0.3908	0.9989	0.9322	0.7057	0.5151	
MBG	0.0300	0.0406	coef	0.0158	-0.0144	0.0013	-0.0136	-0.0234	0.0090	0.0145	0.0336	-0.0048	2.10%
			p-val	0.0003***	0.0138**	0.8908	0.1840	0.1046	0.6432	0.086*	0.0002***	0.7248	

Table 2.5: Fama-French

Risk free rate of 0.01 means a daily risk free rate of 1 basis point, corresponding to an annualized rate of $(1+1/10000)^{252}-1 = 0.0229229$ basis point. *** means significant at 1% level, ** means significant at 5% level, * means significant at 10% level, the excess return being used here is the portfolio daily excess return on committed capital.

In Table 2.5, we report the R-squared for regression, coefficient, and p-value for fac-

tors and residual (alpha). Moreover, we report the annualized alpha for easier comparison with annualized excess return on committed capital.

The R-squared of all three sets of factors indicates that the profit of all pair trading strategies are not fully explained by these factors (all R-squared values are less than 5%). The annualized alphas of all strategies are proportional to their annualized excess return on committed capital. It is worth mentioning that the alphas for the cointegration and MBG methods are both positive and significant. For Fama–French 3 research factors, we record significant loading for the value factor (HML) for the distance, MBG, and cointegration methods, which record a significant level of 10%, 5%, and 1%, respectively, after applying the transaction cost. For the Fama–French 5 research factor, the HML loading for the distance method is no longer significant compared to the Fama–French 3 research factor. The new profitability factor (RMW) is significant at a 5% level for both the distance and MBG methods. For the third set of factors, it is worth noting that the market factor (MKT) becomes significant for the MBG method at a 5% level, and the new short-term reversion (ST-Rev) factor is significant at 1% level. Moreover, for both the cointegration and MBG methods, the momentum factors are significant.

2.4.8 Sensitivity analysis to hyperparameters

In our implementation of pair trading strategies, we fix the enter and exit threshold equal to $+2$ or -2 times the standard deviation of the spread for the distance and cointegration methods. As for the copula method, we fix the threshold to enter a trade when the cumulative mispricing index of one stock exceeds 0.5, and the cumulative mispricing

index of other stocks is smaller than -0.5 . We analyze the sensitivity of the pair trading strategies outcome when these thresholds change.

Figure 2.6 and 2.7 show how various opening thresholds affect the number of trading opportunities of the distance, cointegration and copula methods. We can see from Figure 2.6 that in general, the distance method has the least trading opportunities and the copula method has the highest trading opportunities. As the opening threshold increases, we observe a decrease in the proportion of pairs that get traded per selected pair and a decrease in the average number of trades per selected pair. Figure 2.8 analyzes the sensitivity of convergence of trades to the opening threshold. As the opening threshold increases, the percentage of converged trades decreases, and the average number of days for each trade to converge increases.



Figure 2.6: Average number of trades per selected pair

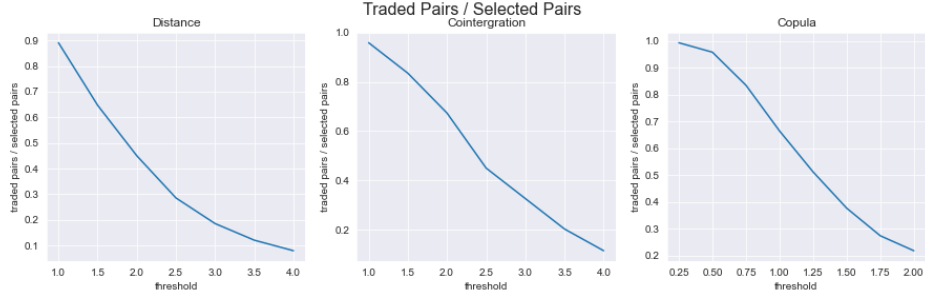


Figure 2.7: Traded pairs / Selected pairs

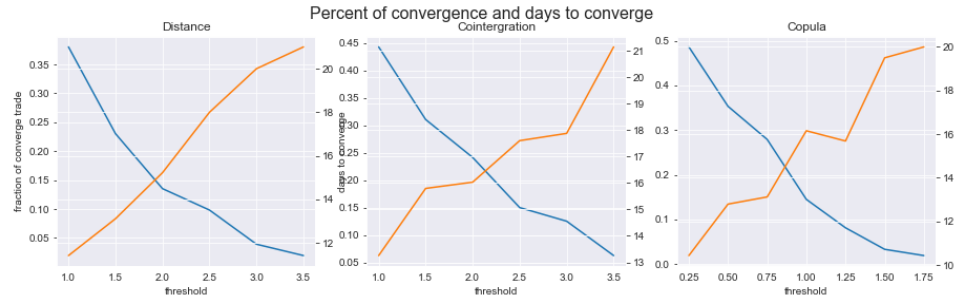


Figure 2.8: Percent of convergence and days to converge

For the MBG method, there are two hyperparameters we could tune instead of only one for the other three pair trading strategies. The first parameter is the lookback period for the cumulative mispricing index, and the second parameter is the threshold for entering a trade. Instead, of fixing the lookback period to 20 days and the threshold of cumulative mispricing index to 0.5, we study the sensitivity of trading opportunities and the robustness of the return by changing these two hyperparameters.

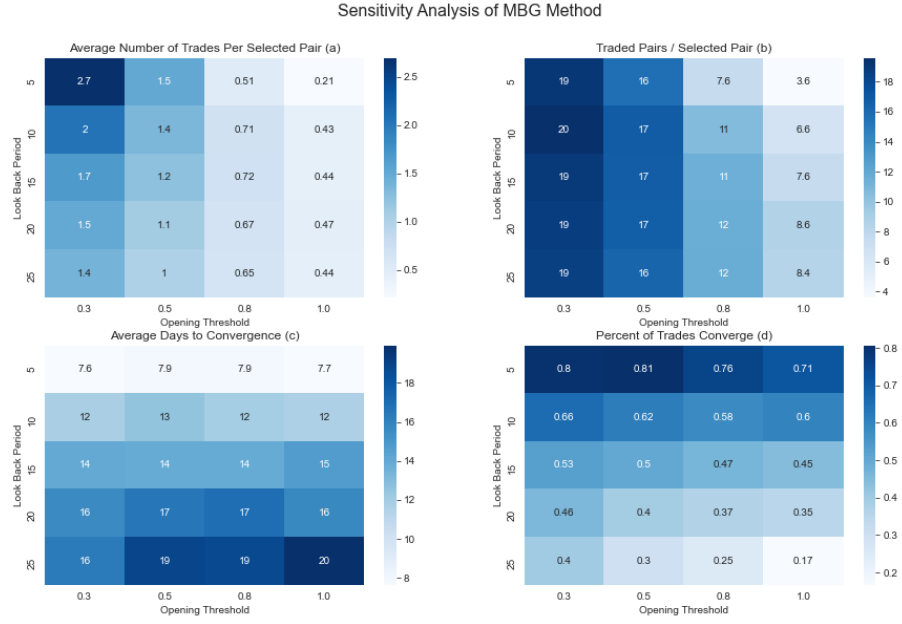


Figure 2.9: Sensitivity analysis of MBG method

From Figures 2.9 (a) and Figure 2.9 (b), we observe that the average number of trades per selected pair as well as the ratio of traded pairs to selected pairs decrease when the lookback period decreases or the opening threshold increases. Figure 2.9 (c) shows that the average number of days for the convergence of each trade increases when the lookback period increase or the opening threshold increase. Figure 2.9 (d) shows that the average percentage of trades that converge increase when the lookback period decrease or the opening threshold decrease.

2.5 Conclusion and future research

A pair trading strategy, categorized as a statistical arbitrage strategy, is a market-neutral strategy that constructs a portfolio by matching long positions with short ones and is designed to profit from all market conditions. This investment strategy identifies and monitors stock pairs with high historical comovement. It opens positions if the usually balanced relationship between two stocks diverges and closes positions when the relationship is restored.

In this paper, we propose a novel approach for designing a pair trading strategy—the MBG method. To the best of our knowledge, this is the first literature to discuss this approach for pair trading. This method uses BG distribution to fit the marginal distribution of stock returns, and this enables us to model the negative skewness, positive kurtosis, and asymmetric movement of returns. With the BG parameters, we calibrate the MBG model to describe the dependency structure of stocks. We further define mispricing measure, which leads to trading signal, and devise a threshold-based trading rule for determining the timing to systematically enter and exit positions. Our finding shows that among the three popular pair trading strategies and our newly proposed method, the MBG method has the highest excess return and records the best risk-adjusted-performance. In the dataset we tested, MBG reported an annualized return on committed capital of 2.4% on average after applying transaction cost, compared to 1.6%, -1.0% , and -1.2% for cointegration, copula, and distance methods, respectively. The MBG method also records a 1.35 annualized Sharpe ratio after applying transaction cost, which is the highest, followed by 1.08, -0.378 , and -1.359 from the cointegration, copula, and distance methods,

respectively. The same ranking of performance is also consistent with other risk-adjusted measures related to upper/lower partial moments, including Omega, Sortino, and Kappa-3 ratios. We also analyze the overall trading opportunities of each strategy and the convergence of trades. The MBG and copula methods record more than 400 opening trades while the distance and cointegration methods record only 200 opening trades on average. As for trade convergence, the distance method has the least convergence rate of 13.4%, and the MBG method has the highest convergence rate of 45.2%. Although the distance and cointegration methods have better returns per converged trade because they have fewer trading opportunities and lower convergence rates, the MBG method has the best overall performance. From the result of the regression against common equity risk factors, including the Fama–French-5 research factors, we record low R-squared (at most 5%) for all four strategies, which indicates that these factors cannot fully explain the excess return for these strategies. Moreover, the robustness of the excess return for the MBG and the cointegration methods is justified by the positive and significant alpha in the result. In conclusion, pair trading continues to be a robust strategy after 40 years of development. As the profitability of the most well-known distance method has declined in recent years, the newly proposed MBG and the cointegration methods remain profitable.

Some potential research topics have been left for the future, and there are some ideas that we would like to explore. In the copula method, instead of using BG marginal and Gaussian copula, it would be interesting to know the performance of different marginal distributions and copula. Potential copulas to test are Clayton, Gumbel, and Student-t copula. Moreover, the dimension of the strategy could be extended to higher dimensions for the MBG method. One could consider changing the entering signal as one asset's

cumulative mispricing index is higher than the upper threshold while the other assets' cumulative mispricing indexes are smaller than the lower threshold.

2.6 Appendix

<i>Strategy</i>	<i>test statistics</i>	<i>p-val</i>	<i>Strategy</i>	<i>test statistics</i>	<i>p-val</i>
Section 1 :Return on committed capital			Section 2 :Return on fully invested capital		
<i>Panel A: After transaction cost</i>			<i>Panel A: After transaction cost</i>		
Distance	0.953	0.514	Distance	0.965	0.734
Cointegration	0.876	0.041	Cointegration	0.847	0.016
Copula	0.955	0.534	Copula	0.962	0.660
MBG	0.936	0.270	MBG	0.968	0.784
<i>Panel B: Before transaction cost</i>			<i>Panel B: Before transaction cost</i>		
Distance	0.956	0.551	Distance	0.965	0.731
Cointegration	0.905	0.113	Cointegration	0.853	0.019
Copula	0.964	0.706	Copula	0.959	0.613
MBG	0.914	0.115	MBG	0.938	0.292

Table 2.6: Shapiro–Wilk test for portfolio excess return

Chapter 3: Predictability of Corporate Bond Returns with Machine Learning

3.1 Introduction

A fundamental question in the field of finance is the estimation of the relationship between the characteristics, risk exposure, and expected returns on assets. There has been extensive study over the past decades in the predictability of asset returns to answer this question. There are three strands of literature on the prediction of empirical asset returns. The first group of literature models the expected returns of different assets using asset-level characteristics. Portfolios based on different characteristics are created and sorted. The typical approach involves running time series regression, which was pioneered by [Fama and French \(1993\)](#). The second approach—different from the first—usually performs cross-sectional regression on several macroeconomic predictors instead of returns on portfolios with different characteristics. The classical literature was surveyed by [Subrahmanyam \(2010\)](#) and [Goyal \(2012\)](#). The third group using machine learning methods and dimension reduction techniques has recently emerged. This approach triumphs as the dimension of factors and characteristics increased dramatically and as academics discovered them over the past decades. In the application of machine learning methods in

asset returns, [Rapach et al. \(2013\)](#) applied the least absolute shrinkage and selection operator (Lasso) to investigate the lead–lag relationship among the monthly stock returns of different countries. [Khandani et al. \(2010\)](#) and [Butaru et al. \(2016\)](#) used decision tree and random forest to predict credit card delinquency. More recently, [Gu et al. \(2020\)](#) surveyed a wide range of machine learning methods, including random forest, boosted regression tree, and deep neural network for US stock return prediction. In that paper, the authors illustrate that the machine learning methods have made a substantial improvement when forecasting expected returns. Moreover, the portfolios constructed by using machine learning prediction display a significantly superior return compared to the benchmark portfolio.

In the empirical asset pricing literature, more works have been focusing on the equity market than on fixed income, especially in corporate bonds. We believe the main reason behind this is the lack of easy access to the source of reliable data. The pioneering work in forecasting bond returns was done by [Fama and French \(1993\)](#) , who related default spread and term spread to bond market returns. [Lin et al. \(2011\)](#) discovered that there is a positive relationship between expected corporate bond returns and liquidity beta. [Bai et al. \(2016\)](#) found that bond-level distributional characteristics have significant prediction power on bond returns.

In recent years, with the gain in popularity of machine learning, increase in trading volume of corporate bonds in the financial market, and the availability of corporate bond data, academics have begun applying machine learning methods in predicting corporate bonds returns. To the best of our knowledge, only a handful of papers have reported research on such topics. The first work was done by [Bali et al. \(2020\)](#) where the authors

surveyed a handful of machine learning methods and tested their predictability on corporate bond returns. Following [Bali et al. \(2020\)](#), [He et al. \(2021\)](#) extended the study by extending the testing period to 1976 and adding 20 macroeconomic predictors in the forecasting task. [Kim et al. \(2021\)](#) utilized a machine learning method to predict credit spread instead of bond returns.

In this paper, we provide a comparative study on the cross-sectional prediction of corporate bond returns using a large number of corporate bond characteristics. First, we construct our 20+ bond-level features, including bond-level, downside risk, systematic risk, bond-level liquidity, bond level momentum, and distributional characteristics. We study the predictability of corporate bond returns using machine learning methods, including linear models such as Ridge and Lasso, and nonlinear models including decision tree, random forest, boosting tree, and neural network. The traditional linear regression model, such as OLS, produces an out-of-sample R-squared value of 4.10%, while XGBoost model—the machine learning method that performs best in our dataset—has an out-of-sample R-squared value of 5.59%.

In [Bali et al. \(2020\)](#), a similar study has been conducted. Their best model is a neural network that yields a 3.99% out-of-sample R-squared value. By updating their architecture, our neural network results in an out-of-sample R-squared value of 5.06%. Another difference is that we use cross-validation for hyperparameter tuning. In prior research, the whole dataset was divided into three disjoint parts for training, validating, and testing. For example, from the study by [Bali et al. \(2020\)](#), the training period was 3 years, the validation period was 2 years following the end of the training period, and the most recent 10 years data were used for testing. This method is easy to implement;

however, the most significant drawback is the fact that we rely on only one validation error to select hyperparameters. We have tried this method initially but quickly realized that it yields an unstable result, especially for complex models. In our paper, we use the walk forward cross-validation method by [Hyndman and Athanasopoulos \(2018\)](#) to perform cross-validation for hyperparameter tuning while maintaining the temporal order of the data.

To understand which variable has the most predictive power, we then analyze the importance of the predictors in our models using three different approaches. The first approach is to rely on software packages' default options in each model. Although this model is the easiest to implement, its biggest drawback is the lack of consistency across different models. The second approach is the reduction in the R-squared method. This method is commonly used in almost all relevant studies; however, the drawback is the fact that it is not consistent, which means that if a model changes, the marginal contribution of a feature increases, and the feature importance calculated by the reduction in R-squared method might decrease instead. Furthermore, this method does not consider the interaction between features. We also test this method in our dataset, and it leads to erroneous results for random forest and XGBoost as it fails to consider the interaction among features. The third method uses Shapley value and could solve the two drawbacks in the second method. In our research, we use SHAP to measure feature importance. SHAP is an approach that has its root in game theory and is based on Shapley value. In this method, each feature acts as a player in a coalition game; therefore, it takes the interaction between features into account. Moreover, it is consistent, which means an increase in the marginal contribution of feature leads to an increase in Shapley value. Furthermore, SHAP can

provide an explanation for each observation where reduction in R-squared cannot. An explanation for each observation is critical in trading; with this advantage, we would have more confidence in making each trading decision when we use machine learning models.

To estimate the profitability of the machine learning prediction models, we take a deeper dive into the machine learning portfolio construction. For each machine learning model, we construct long-only, short-only, and long-short trading strategies, and there are a total of 25 strategies, including the buy-and-hold benchmark strategies. On average, the long-short, long-only, and short-only strategies record an annualized excess return of 9.60%, 12.05%, and -2.65% , respectively, and an annualized Sharpe ratio of 1.02, 1.21, and -0.28 respectively. Therefore, it is fair to say that long-only strategies have the best performance on average. Among the 25 strategies, the best performing one is the long-only portfolio by XGBoost, which displays an annualized excess return of 13.93% and a Sharpe ratio of 1.31. In previous literature, the model is only trained once during the training period, and the portfolio is rebalanced each month; however, the model was never updated throughout the testing period. (20 years by [Gu et al. \(2020\)](#) and 10 years by [Bali et al. \(2020\)](#)). Using the same model for multiple years is not realistic as in a real trading environment; practitioners usually re-calibrate after a certain timeframe to take advantage of the most recent historical data. Our paper takes a more practical approach as we retrain our model each month-end and only use the prior three months of data for calibration.

One of the biggest challenges in dealing with corporate bond data is the missing value. When using traditional regression models like OLS, researchers often end up in a dilemma scenario such that a significant chunk of data contains missing values. One could either perform missing value imputation but introduce noise into the dataset, or remove all

the records containing missing values; however, this would decrease the amount of data that is already scarce. In our research, we propose the use of the XGBoost model, which is an implementation of the boosting tree, and it uses the sparsity-aware split finding algorithm (Chen and Guestrin, 2016) to perform learning on observations that contain missing information. We test this idea on our dataset by training two boosting models, one with missing values and another without missing values. The boosting model that contains missing values outperforms the other model when only using bond characteristics.

Lastly, we built parsimonious machine learning models with very few features that we deem important and analyze their behavior and model performance shrinkage. The reason for doing this is because, in an actual trading environment, it is always much cheaper and more operationally effective to maintain a data pipeline of a small set of features.

The rest of this paper is structured as follows. In Section 2, we introduce the machine learning models that we use, our performance evaluation metrics, and feature importance metrics. In Section 3, we introduce our dataset and the methodology for bond characteristics. A study on US corporate bond return prediction is discussed in Section 4; the feature importance is presented in Section 5. In Section 6, we test the profitability of bond return predictions through three trading strategies by constructing machine learning portfolios. In Section 7, we test the performance of parsimonious machine learning models that only contain features with top feature importance. We conclude our study in Section 8.

3.2 Research Method

3.2.1 Methodology

This section describes the machine learning methods used in this paper. A new machine learning model will be introduced in each subsection. Previous literature has studied these models well; therefore, our focus here will be their connection with risk premium predictions. Using the prediction error model, the excess return on asset i at time $t + 1$, $r_{i,t+1}$ could be expressed as:

$$r_{i,t+1} = E_t(r_{i,t+1}) + \varepsilon_{i,t+1}$$

where

$$E_t(r_{i,t+1}) = g(z_{i,t})$$

is the time- t expected return for asset i and $g(\cdot)$ is the function that accepts $z_{i,t}$, asset i 's characteristics at time t as input. Assets are indexed by $i = 1, \dots, N$, and months are indexed by $t = 1, \dots, T$, where N is the number of assets at time t . The objective is to search for a function g to represent $E_t(r_{i,t+1})$ that maximizes the out-of-sample explanation power for realized return $r_{i,t+1}$.

3.2.1.1 Linear regression

The first model we describe is the linear regression prediction estimated using OLS. This simple model is the most widely-used method in the financial industry. In this model,

we restrict our approximation of $g(\cdot)$ to a linear function such that:

$$g(z_{i,t}; \theta) = z'_{i,t} \theta$$

where our model parameters θ is estimated by the ordinary least squares by solving the following minimization problem:

$$\min_{\theta} \mathcal{L}(\theta) \equiv \frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T (r_{i,t+1} - g(z_{i,t}; \theta))^2$$

3.2.1.2 Penalized linear regression: Lasso, Ridge, and Elastic net

One of the most commonly used methods to reduce the overfitting issue of ordinary least square is adding a penalty term to the least square objective function. The penalty is introduced as a trade-off between the in-sample performance and its out-of-sample stability. The objective function to minimize to find θ is

$$\min_{\theta} \mathcal{L}(\theta; \cdot) \equiv \mathcal{L}(\theta) + \phi(\theta; \cdot)$$

where $\phi(\theta; \cdot)$ denote the penalty term. By adding this additional term, the estimated θ could be regularized and shrunk toward zero.

In machine learning literature, a popular penalty function is

$$\phi(\theta; \lambda, \rho) = \lambda(1 - \rho) \sum_{j=1}^P |\theta_j| + \frac{1}{2} \lambda \rho \sum_{j=1}^P \theta_j^2$$

where $\lambda > 0$ is a hyperparameter for controlling the degree of regularization. The amount of regularization increases as we increase the value of λ . For this specific $\phi(\theta; \cdot)$, $\lambda = 0$ will reduce the estimated parameter to the standard OLS estimate. When $\rho = 0$, this model becomes the Lasso model, under which a subset of θ may shrink to exactly zero, and because of this property, the Lasso model is a modeling technique and can also be used for variable selection. When $\rho = 1$, this model becomes Ridge Regression, which shrinks parameters closer to zero but does not precisely zero as in Lasso. Therefore, Ridge Regression is a dense modeling technique for controlling the magnitude of the parameters. If ρ is between 0 and 1, the model becomes the elastic net model, representing a mixture between the Ridge and Lasso. Similar to Ridge and Lasso, one advantage of the elastic net is that it can handle highly correlated characteristics ([Diebold and Shin, 2019](#)).

3.2.1.3 Regression tree

Tree-based models like the regression tree are built in a sequence of steps. In each step, a new branch separates the data into two smaller subsets based on one predictor. Conceptually, a tree partitions the feature space into a set of small rectangles. Each partition is called a leaf node, and this final output is the average values of returns in each partition. Figure [3.1](#) is an example of a regression tree with two predictors, “Rating” and “Maturity”. This tree partitions the two-dimensional feature space into three regions. Firstly, observations are partitioned based on their time to maturity. The observations with maturity larger than or equal to 20 years are assigned to region R_3 ; those with maturity less than or equal to 10 years are further divided by rating. Bonds with a rating higher

than or equal to 10 are assigned to region R_2 ; while the rest go into region R_1 .

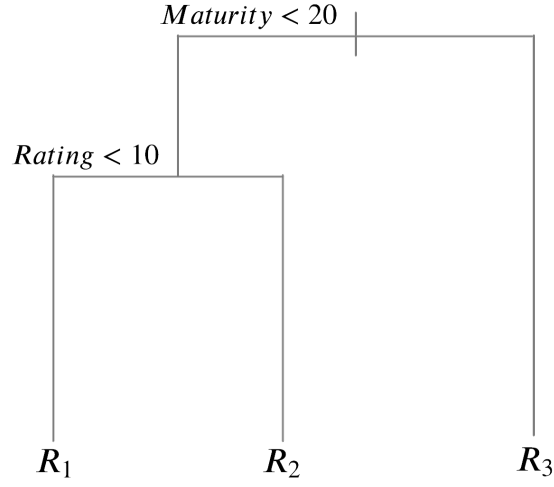


Figure 3.1: Regression tree example

Mathematically, we can express the expected return function $g(\cdot)$ as:

$$g(z_{i,t}; c, K, L) = \sum_{k=1}^K c_k 1_{\{z_{i,t} \in R_k\}}$$

and c_k is the average of outcomes for observations in the k -th partition

$$c_m = \text{Avg}(y_i \mid x_i \in R_m)$$

where K is the number of leaf nodes in the tree, and L is its height, R_k is one of the K partitions of the data, $1_{\{\cdot\}}$ is the indicator function. The prediction equation in Figure 3.1

is:

$$g(z_{i,t}; c, 3, 2) = c_1 1_{\{\text{maturity}_{i,t} < 20\}} 1_{\{\text{rating} < 10\}} + c_2 1_{\{\text{maturity}_{i,t} < 20\}} 1_{\{\text{rating} \geq 10\}} + c_3 1_{\{\text{maturity}_{i,t} \geq 20\}}$$

We use the CART algorithm (Breiman et al., 1984) to decide our splitting criteria in the formula above. In this algorithm, one predictor value pair is chosen in each step to perform split region into two. The objective is to maximize the discrepancy (impurity) between the average returns in each subregion after the split. A single tree could be subject to a potential overfitting problem; therefore, in our study, we limit the maximum depth of the tree as a regularization mechanism.

3.2.1.4 Random forest

Another way of adding regularization to tree-based algorithms is to use an ensemble learning method called Bagging (bootstrap aggregation). This method is commonly used to reduce variation within a dataset, and the general procedure involves creating random samples of data with replacement in the training set. In random forest, as the name “forest” suggested, we build more than one tree, and each tree is trained using one batch of samples drawn from full training data with replacement. Also, to further decorrelate the tree, in each branch split, only a subset of features is considered.

To grow a random forest with B trees, the general steps are described below. For b from 1 to B , we first generate bootstrap samples $\{(z_{i,t}, r_{i,t+1}), (i, t) \in Z_b\}$ where Z_b is the b -ith bootstrap samples from the original dataset. To grow a tree in a random forest, we recursively repeat the following steps: 1) Select m predictors at random from the total

of P variables. 2) Pick the best variable/split-point among the m predictors, 3) Split the current node into two child nodes. The result of the forest is:

$$g(z_{i,t}) = \sum_{b=1}^B T_b(z_{i,t})$$

where T_b is the prediction given by the b -th regression tree in the forest.

3.2.1.5 Gradient boosting

In contrast with random forest, which is an ensemble model that consists of deep trees (strong learners), boosting is a procedure that combines the outputs of many shallow trees (weak learners) to produce a powerful “committee.” The theory behind boosting suggests that an ensemble of many oversimplified trees could outperform a single compare tree and have greater stability. There are different boosting trees, and the one used in this study is called gradient boosted regression trees. The general algorithm is described in Algorithm [3.2.1](#):

Algorithm 3.2.1 Gradient Boosting

Initialized the predictor as $\hat{g}_0(\cdot) = 0$

for b from $1, 2, \dots, B$ **do**

 compute for each $i = 1, 2, \dots, N$ and $t = 1, 2, \dots, T$, the negative gradient of the loss function $L(\cdot, \cdot)$. For regression problem, we use l_2 loss here so the negative gradient is $(y_{i,t} - g)$

$$\varepsilon_{i,t+1} \leftarrow - \left. \frac{\partial L(r_{i,t+1}, g)}{\partial g} \right|_{g=\hat{g}_{b-1}(z_{i,t})}$$

 Grow a regression tree of depth D with dataset $\{(z_{i,t}, \varepsilon_{i,t+1}) : \forall i, \forall t\}$

$$\hat{f}_b(\cdot) \leftarrow g(z_{i,t}; \theta, D)$$

 Update the prediction by

$$\hat{g}_b(\cdot) \leftarrow \hat{g}_{b-1}(\cdot) + \nu \hat{f}_b(\cdot)$$

where $\nu \in (0, 1]$ is a hyperparameter called learning rate that controls the step length.

end for

Result: This output of the model is

$$\hat{g}_B(z_{i,t}; B, \nu, D) = \sum_{b=1}^B \nu \hat{f}_b(\cdot).$$

3.2.1.6 Feed-forward neural network

Neural networks are a collection of algorithms to model relationships in data through a process that mimics how the human brain operates. A typical neural network consists of an input layer that contains the input, hidden layers that interact and transform the predictors, and an output layer that provides final prediction. For a feed-forward neural network (FFN), the information moves only in forward directions through the hidden layers to the output layer, and no cycles or loops exist. For an FFN with L hidden layers, the model could be represented as

$$G(z, b, W) = b^{L-1} + (z^{L-1})' W^{L-1}$$

$$z_K^l = g \left(b^{l-1} + (z^{l-1})' W^{l-1} \right) \quad l = 1, \dots, L-1$$

$$z^0 = \{z_{i,t} : \forall i, \forall t\}$$

where z^0 is our input data, $g(z_k) = \max(z_k, 0)$ is the rectified linear unit activation function, and $G(\cdot)$ is the final output. The model parameters are W and b , called weights and biases, respectively.

3.2.2 Training, validation, and test sets

The most common approach for splitting data in relative literature is to divide the full sample into three disjoint time period—training, validation and testing set. In this approach, after the model is trained on training dataset, hyperparameters are selected

based on the validation set, and the final performance is evaluated by testing against the test set. One drawback for this approach is that we rely only on result from the evaluation on our validation set, and this might introduce biases in our model. Another shortcoming of this approach is that we need to set aside a large chunk of data and cannot use it in model training, thereby wasting data. A better approach is to use cross-validation instead of validation. However, when dealing with time series data, the traditional k -fold cross-validation does not work, and a popular alternative is the walk forward cross validation (Hyndman and Athanasopoulos, 2018). Under this setting, we first split our data into two parts—the training and testing sets—then within the training set, we further generate a series of validation and training sets and name them validation sets for cross-validation and training sets for cross-validation, respectively. The series of validation sets for cross-validation are nonoverlapping and their respective training sets for cross-validation consist only of observations that occurred prior to the observations that forms the test sets. It is worth noting that for each fold of the cross-validation, there is a time period of size k between the end of the training period and the start of the testing period. This period is called the lag period and the reason for it is because we use information up to $t - k$ when we calculate some features for time t . If we fail to impose such gap, the validation set will contain information from the training set implicitly (from the features) and this will underestimate the validation error. In this method, there are three parameters that need to be set, the size of the validation sets for cross-validation, the minimum size of the training set for cross-validation, and the lag period.

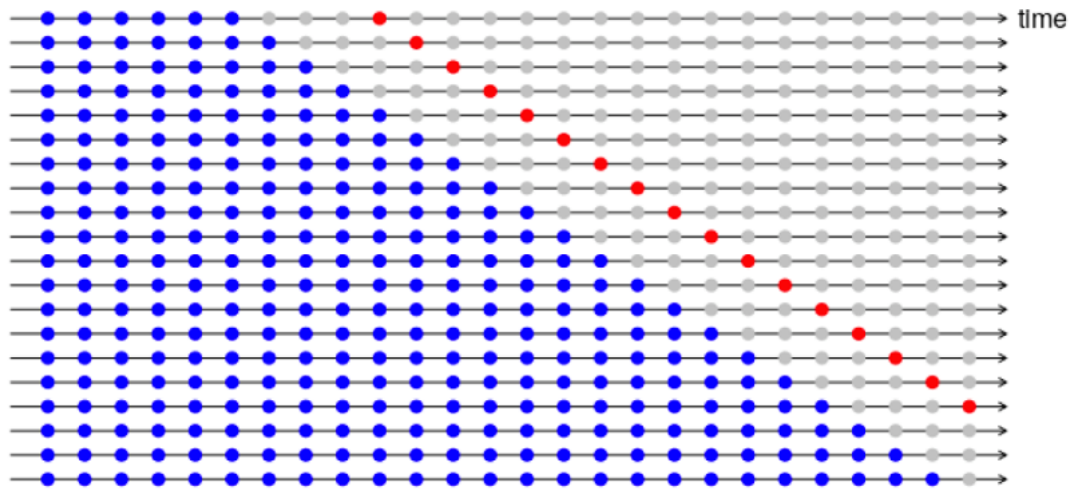


Figure 3.2: Walk forward cross validation

The graph above illustrates this idea. The red dot on each line represents validation sets for cross-validation and the set of blue dots represents the training sets for cross-validation. In our actual dataset, our training data is from 2002 to 2016 and our testing data is from 2017 to 2020. The minimum length of the training sets for cross-validation is 4 years, and the size of each validation set for cross-validation is 2 years.

3.2.3 Performance evaluation metric

3.2.3.1 Out-of-sample R-squared

Our paper adopts the prevailing out-of-sample R-squared as our performance evaluation metric. Out-of-sample R-squared is a popular way of assessing the model performance in cross-sectional performance literature, and it was first reported by [Campbell and](#)

Thompson (2008) with definition.

$$R_{OS}^2 = 1 - \frac{\sum_{t=1}^T (r_t - \hat{r}_t)^2}{\sum_{t=1}^T (r_t - \bar{r}_t)^2}$$

The R-squared is evaluated on test data only, where r_t is the actual return, $\hat{r}_t$ is the model prediction, and $\bar{r}_t$ is the prediction of the null model, that is, the historical average return.

In our research, instead of using historical mean return, we use zero as the null model prediction as suggested by Gu et al. (2020) because unlike portfolio and index level return, asset-level return is so noisy; therefore, predicting future returns with historical average underperforms a forecast of zero. The R_{os}^2 is defined as,

$$R_{\text{os}}^2 = 1 - \frac{\sum_{(i,t) \in \mathcal{T}} (r_{i,t+1} - \hat{r}_{i,t+1})^2}{\sum_{(i,t) \in \mathcal{T}} r_{i,t+1}^2}$$

where $\mathcal{T}$ is the testing set.

Unlike the traditional R-squared, the out-of-sample R-squared ranges from negative infinity to 1. A negative value suggests that the model's performance is worse than the benchmark, whereas a positive value indicates that the model outperforms the benchmark.

3.2.4 SHapley Additive exPlanations (SHAP)

It is critical to understand the rationale behind how a model makes such prediction and to correctly interpret a predictive model's output. These would increase user trust, provide suggestions for model improvement, and enhance the understanding of the model

decision-making process. In the field of finance, simpler models such as linear regression are often preferred since it is easier to interpret compared with a more complex model, even though the latter could be more accurate. However, to train high accuracy models when facing large modern datasets, it is almost inevitable that we end up with complex models that are hard to interpret, such as deep learning models. This situation creates a trade-off between the accuracy and interpretability of the model. To address this problem, [Lundberg and Lee \(2017\)](#) presents a unified framework for interpreting predictions called SHAP(SHapley Additive exPlanations). For each prediction, SHAP assigns an importance score for each feature. Moreover, it identifies a new class of additive feature importance measures, and the theoretical results show the solution is unique with a set of desirable properties.

SHAP's approach to interpreting predictions is based on game theory which uses the classical Shapley values with related extensions. Shapley value, named in honor of Lloyd Shapley, who introduced it in 1951 and won the Nobel prize for it in 2012, is a solution concept in cooperative game theory.

SHAP retrain model on all feature subsets $S \subseteq F$, where F is the set of all features. After retraining, an importance value is assigned to each feature, and it indicates the effect of the model prediction for including such a feature. To compute this value, we first train a model $f_{S \cup \{i\}}$ with such feature present, then train another model f_S with the feature excluded. Then, the predictions from the two models are compared on the current input $f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)$, where x_S is the values of the features in S . Subsequently, we proceed with computing the difference for all possible subset $S \subset F \setminus \{i\}$. Finally, Shapley values are computed as a weighted average of all possible differences and used

as feature attributions.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]. \quad (3.1)$$

An attribute of the class of additive feature attribution methods is the existence of a unique solution with three desirable properties.

The first property is local accuracy. If we try to approximate the model f at x , the local accuracy property requires the explanation model should equal to the output of f for the simplified input, x' .

The second property is missingness. If the simplified inputs represent feature presence, then missingness requires features missing in the original input to have no impact. All of the methods described in Section 2 obey the missingness property.

Property 1. **Local accuracy**

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^M \phi_i x'_i$$

The explanation model $g(x')$ is equal to the original model $f(x)$ when $x = h_x(x')$

Property 2. **Missingness**

$$x'_i = 0 \implies \phi_i = 0$$

Missingness constrains features where $x'_i = 0$ to have no attributed impact.

Property 3. **Consistency**

The third property is consistency. Consistency means that if a model output changes such that some simplified input's contribution increases or stays the same. Then regardless of the other inputs, such input's attribution should not be decreased.

Let $f_x(z') = f(h_x(z'))$ and $z' \setminus i$ denote setting $z'_i = 0$. For any two models f and f' , if

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)$$

for all inputs $z' \in \{0, 1\}^M$, then $\phi_i(f', x) \geq \phi_i(f, x)$

Definition 3.1. There is only one explanation model, g , which follows the definition of Shapley value in Equation 3.1 and satisfies Properties 1,2, and 3 at the same time:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|! (M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

where $|z'|$ is the number of non-zero entries in z' . $z' \subseteq x'$ represents all z' vectors where the nonzero entries are a subset of the non-zero entries in x' .

Under these 3 properties, for any simplified input mapping h_x , Definition 3.1 shows that the additive feature attribution method is unique and it result implies that methods not based on Shapley values violate local accuracy and/or consistency.

3.3 Data and Variable Definitions

3.3.1 Corporate bond data

3.3.1.1 TRACE

Trade Reporting and Compliance Engine (TRACE) is the over-the-counter real-time price dissemination service for the fixed income market from the Financial Industry Regulatory Authority (FINRA), which brings transparency to corporate and agency bond markets ([tra, 2021](#)). TRACE contains consolidated transaction-level data such as price, the date and time of execution, transaction size, and the yield. Additionally, it contains information such as buy/sell indicators and counterparty (dealer or customer) information. We use two different TRACE data packages in our research: TRACE Standard and TRACE Enhanced. These two are very similar, with three main differences. The first difference is that standard TRACE has a transaction size cap while enhanced TRACE does not. The transaction cap is 5 million for investment-grade (IG) bonds and 1 million for high-yield (HY) bonds. For example, with this cap, any transaction of IG bond with more than 5 million will show up as 5 million in the TRACE record. Secondly, standard TRACE has a phased launching scheme. It starts with only IG bonds with more than 1 billion original issue size in July 2002 to all IG and HY bonds after Sep 2004. Thirdly, unlike standard TRACE, which is real-time, enhanced TRACE has 18 months of delay. Therefore, if one needs the most recent transactions, TRACE Enhanced is not sufficient, although it contains more information than standard TRACE.

In our research, the TRACE dataset contains data from July 2002 to September

2020. We use TRACE Enhanced whenever possible to get the most accurate transaction data. For the most recent time from March 2019 to September 2020, when TRACE Enhanced was unavailable, we substitute with TRACE Standard.

Since TRACE is a self-reported tool used by market participants, reporting mistakes are anticipated in the data, and subsequent correction is allowed by the FINRA. The three most common methods of revising previous errors are cancellation (void a previous same-day trade report), correction (void a previous same-day trade report and replace with updated information), and reversal (void a previous trade report and replace with updated information). To get the correct information, one needs to clear out these results. Also, there will be a trade report for the same trade on both sides (both buy and sell) for the dealer-to-dealer transaction, which leads to “double counting,” and one needs to address that. To get to the clean TRACE data, we leverage on the SAS code written by (Freda) Song Drechsler in 2017 ([Drechsler, 2017a,b](#)), which is available in Wharton Research Data Services (WRDS) and [Dick-Nielsen \(2009, 2014\)](#), who discussed the data cleaning process of TRACE in detail.

3.3.1.2 Mergent FISD data

While transaction-related bond data are provided by TRACE, Mergent Fixed Income Securities Database(FISD) database contains reference data on a bond level like issue date, maturity date, coupon rate, coupon frequency, bond rating, and default information since January 1950. For our research, the bond-level nontransaction data comes

from FISD, which is available from WRDS.

3.3.1.3 WRDS bond return data

The WRDS Bond return database ([wrd, 2021](#)) is a novel and unique corporate bond database compiled by WRDS researchers using TRACE and FISD data feed. This database provides easy access to cleaned datasets of corporate bond transactions along with monthly price and returns information. In our research, we use this database for monthly returns and price data and other transaction-level and reference data whenever it is provided. However, because this database has limited fields and a monthly record frequency, we still need to use the TRACE dataset directly to access intraday and daily transaction-level data.

3.3.1.4 Other data, Bloomberg Barclays index and Fed H.15 CMT yield

Bloomberg Barclays indices have been the market standard for fixed-income investors seeking objective, rule-based, and representative benchmarks for measuring asset classes risk and return since 1973. Specifically, the Bloomberg Barclays US Corporate Total Return Value Unhedged USD (LUACTRUU) and the Bloomberg Barclays US Corporate High Yield Total Return Value Unhedged USD (LF98TRUU), ([Bloomberg, 2021](#)) are the two indexes we later use in our research to approximate the corporate bond market portfolio returns. Another data set we use is constant maturity treasury yield data within

the United States Federal Reserve Statistical Release H.15, which is a weekly publication (with daily updates) of the Federal Reserve System of selected market interest rates ([fed, 2022](#)). We will use this data later in the research to approximate the TERM factor.

3.3.2 Corporate bond return

The monthly corporate bond return, $R_{i,t}$, for bond i at month t is defined as

$$R_{i,t} = \frac{P_{i,t} + AI_{i,t} + C_{i,t}}{P_{i,t-1} + AI_{i,t-1}} - 1$$

where P stand for clean price, AI stands for accrual interest, and C stands for coupon payment (0 if no coupon payment at month t).

The monthly excess return, $r_{i,t}$, for bond i at month t is defined as

$$r_{i,t} = R_{i,t} - RFM_{i,t}$$

$$RFM_{i,t} = \sqrt[12]{1 + RFA_{i,t}} - 1$$

where RFA is the annualized risk-free rate approximated by constant maturity yield for one-month treasury at month t in the Fed H.15 statistical release. RFM is the monthly risk free rate calculated from RFA .

Corporate bonds bear credit risk, which means they could default prior to reaching maturity. If we treat default returns as missing observations, we will overestimate the bond returns, especially for the HY bond. We follow the methodology by [Cici et al. \(2017\)](#) to compute the default return for all defaulted bonds. Unlike equity, the corporate

bond still trades after default. Before calculating monthly bond return, we search for any post-default price information in the default month and the subsequent month and use such prices as end-of-month prices to calculate monthly returns. Where we are unable to find post-default prices, we use the IG and HY return average as proxies for default month returns. According to [Cici et al. \(2017\)](#), the default month return is -45.6% for IG issues, and -15.8% for HY issues.

To calculate the bond return for month t , we first need the bond prices for months t and $t - 1$. There are three types of month-end prices that are reported in the WRDS Bond Return database: P_{EOM} , P_{LDM} , and P_{L5M} . P_{EOM} uses the last price at which the bond traded in a given month. P_{LDM} uses the price of the last trading day of the month, if available, and is marked missing if the bond did not trade on that day. P_{L5M} uses the last price at which the bond traded in the last five trading days in a given month. With the three types of prices, R_{EOM} , R_{LDM} , and R_{L5M} are calculated accordingly. In the dataset, the most well-populated return measure is R_{EOM} because it only requires the bond traded in that month, and the most restricted measure is R_{LDM} because it requires the bond traded in the last trading day of the month. In our sample period, 85.4%, 32.4%, and 62.5% of data have non empty value for R_{EOM} , R_{LDM} , and R_{L5M} , respectively. In our research, we chose to work with R_{EOM} because it has the least number of empty records, with the assumption that bond prices will not move by a significant amount until they are traded again.

3.3.3 Bond characteristics

We use the following 21 bond characteristics to predict bond returns. These bond characteristics constructed are motivated by the existing studies on the cross-section of corporate bond return literature and can be categorized into six categories, (1) bond-level characteristics, (2) downside risk characteristics, (3) systematic risk characteristics, (4) bond level liquidity characteristics, (5) bond level momentum and short-term reversal, and (6) bond level distributional characteristics.

3.3.3.1 Descriptive characteristics

Credit Rating (RATE): Our bond-level rating information comes from the Mergent FISD bond rating database. For each bond and each month, there are up to three ratings from the three credit rating agencies (Standard & Poor's, Moody's Investors Service, and Fitch Group) available. We use Table 3.1 to map the rating from letter to numeric value. In this methodology, numeric ratings from 1 to 10 are considered as investment grade, and 11 to 22 are considered as high yield. The overall rating is calculated as the average of the three numbers.

Numeric	S&P	Moody	Fitch
1	AAA	AAA	AAA
2	AA+	AA1	AA+
3	AA	AA2	AA
4	AA-	AA3	AA-
5	A+	A1	A+
6	A	A/A2	A
7	A-	A3	A-
8	BBB+	BAA1	BBB+
9	BBB	BAA2	BBB
10	BBB-	BAA3	BBB-
11	BB+	BA1	BB+
12	BB	BA2	BB
13	BB-	BA3	BB-
14	B+	B1	B+
15	B	B2	B
16	B-	B3	B-
17	CCC+	CAA1	CCC+
18	CCC	CAA2	CCC
19	CCC-	CAA3	CCC-
20	CC	CA	CC
21	C	C	C
22	D	NA	D/DD/DDD

Table 3.1: Credit ratings.

Time-to-Maturity (TMT): The time to maturity for a specific bond in month t is calculated as the number of year fractions between the bond maturity date and the date

on which the bond last traded in month t . For simplicity, we assume all bonds have a day count convention of 30/360.

Age (AGE): The age of a specific bond in month t is calculated as the number of year fractions between the bond first issue date of a bond and the date it last traded in month t . For simplicity, we assume all bonds have a day count convention of 30/360.

Duration (DUR): The duration, measured in years, is the weighted average of time until the full cash flows are received.

3.3.3.2 Downside risk characteristics

[Huang et al. \(2012\)](#) suggested that bearing of extreme downside risk (EDR) is rewarded by higher expected future returns in the equity market. In this article, the author also mentions that Value-at-Risk (VaR) plays a significant role in explaining the EDR premium. We use VaR and Expected Shortfall (ES) to approximate the EDR factor in our research.

5% Value-at-Risk (VAR5): Our calculation of the 5% VaR for a bond is based on the left tail of the monthly return distribution. VAR5 is the fifth percentile of the monthly return distribution. For VAR5, we take the monthly return from month t to month $t - 35$ to construct the empirical distribution. If there is no missing data, the second lowest monthly return is the 2.85 percentile, and the third lowest monthly return is the 5.71 percentile. The fifth percentile is calculated as the linear interpolation between the second and the third lowest monthly returns. Moreover, we multiple this result by -1 for the convenience of interpretation.

10% Value-at-Risk (VAR10): Similar to VAR5, to calculate VAR10 for month t , we take the monthly return from month t to month $t - 35$ to construct the empirical distribution. If there is no missing data, the fourth lowest monthly return is the 8.57 percentile, and the fifth lowest monthly return is the 11.4 percentile. The 10th percentile is calculated by linear interpolation between the fourth and fifth lowest monthly returns. Moreover, we multiple this result by -1 for the convenience of interpretation.

5% Expected Shortfall(ES5): Similar to VaR, expected shortfall (ES) is an alternative risk measure. The $x\%$ ES is defined as the expected return on the asset in the worst x percent of cases. To calculate the ES5 for month t , we take the monthly return from month t to month $t - 35$ to construct the empirical distribution and use the average of the worst two monthly returns to approximate the ES5.

10% Expected Shortfall (ES10): The construction of ES10 is similar to ES5 except that we use the average of the worst four monthly returns to approximate this number.

3.3.3.3 Systematic risk characteristics

[Fama and French \(1993\)](#) used three stock market factors (market factor, size factor, and value factor) and two bond market factors (default factor and term factor) to explain the cross-sectional returns of the bond and equity market. In our research, our prediction focuses on the corporate bond market; therefore, so we construct our approximation of market factor, default factor, and term factor.

Bond Market beta (MKT): According to [Fama and French \(1993\)](#), the market factor is designed to capture the excess return on the equity market. It is calculated as the

difference in returns on the value-weighted portfolio of stocks and the 1-month treasury bill rate. In our research, we calculate the bond market factor using a portfolio of corporate bonds instead of stocks. First, we calculate the return of Bloomberg Barclays US Corporate Total Return Value Unhedged USD (LUACTRUU) Index and Bloomberg Barclays US Corporate High Yield Total Return Value Unhedged USD (LF98TRUU) Index ([Bloomberg, 2021](#)) to approximate the return on IG and HY bonds for any given month. Second, from TRACE transaction data, we calculate the dollar volume of all IG bond transactions and all HY bond transactions for each month, standardize the volume, and divide them by the total volume to come up with weight. Third, we then use the returns on the two indexes and their weights to calculate the weighted average return in order to approximate the bond market return for that month. The last step is to calculate the excess return, and we use constant maturity yield for the 1-month treasury bill published in the Federal Reserve H.15 Statistical Release as the proxy for the risk-free rate and subtract that from the bond market return. The excess return calculated above will be our bond market factor. The MKT is then calculated from the time series regression of the excess return on individual bonds against the excess return of the bond market using a 36-month rolling window.

Default Factor beta (DEF): In the first construction, the authors use the difference of return on a market portfolio of long-term corporate bonds and the long-term government bond to calculate the default factor. In our research, the return on the long-term corporate bond is calculated by following the first three steps in the calculation of the MKT, and the return on the long-term government bond is approximated by the return on the Bloomberg Barclays US Treasury Total Return Value Unhedged USD (LUATTRUU)

Index. The default factor is calculated by subtracting the return on the LUATTRUU Index from the return of the long-term corporate bond. With the default factor, the DEF is calculated from the time series regression of excess return on individual bond against the default and bond market factors using a 36-month rolling window.

Term Factor beta (TERM): Originally, the term factor was the difference between the monthly long-term government bond return and the 1-month Treasury bill rate. [Fama and French \(1993\)](#) suggested that using 1-month bill rate is meant to proxy for the general level of expected returns on 1-month treasury bills. By following the same logic, in our research, the term factor is the difference between the return of the Bloomberg Barclays US Long Treasury Total Return Value Unhedged USD (LUTLTRUU) Index and constant maturity yield for the 1-month bill published in the Federal Reserve H.15 Statistical Release. The TERM is then calculated from the time series regression of excess return on individual bond against term factor and bond market factor using a 36-month rolling window.

3.3.3.4 Liquidity characteristics

Illiquidity (ILLIQ): We follow [Bao et al. \(2011\)](#) to construct this illiquidity measure which is based on the magnitude of transitory price movements. We start by assuming the log price of a bond at month t and day d , where $p_{t,d}$ consist of two components: $f_{t,d}$ and $u_{t,d}$, and can be represent by

$$p_{t,d} = f_{t,d} + u_{t,d}$$

where $f_{t,d}$ represents the fundamental value of bond price and $u_{t,d}$ is the transitory price component that comes from the impact of liquidity.

Based on prior work by [Niederhoffer and Osborne \(1966\)](#) and [Vayanos and Wang \(2009\)](#), the liquidity demander's transaction causes the price to deviate from fundamental value which will be corrected in the next time interval. Therefore, price changes exhibit negative auto covariance and the impact will be larger in magnitude if the asset is less liquid, resulting in more price impact. The measure of illiquidity γ is then defined as

$$\gamma = -\text{Cov}(\Delta p_{t,d}, \Delta p_{t,d+1})$$

$$\Delta p_{t,d} = p_{t,d} - p_{t,d-1}$$

Roll's daily measure of illiquidity (ROLL84): [Roll \(1984\)](#) purposed a measure of effective bid-ask spread using asset returns. In our research, we follow the same setup and define the illiquidity measure as

$$\text{Roll} = \begin{cases} 2\sqrt{-\text{cov}(r_d, r_{d-1})} & \text{if } \text{cov}(r_d, r_{d-1}) < 0 \\ 0 & \text{otherwise} \end{cases}$$

Where $r_{t,d}$ is the daily log return on bond at month t , day d . Moreover, where we have less than four observations in a given month for a specific bond, we will mark it as missing.

Fraction of Zero transaction days (LIQZERO): Following [Lesmond et al. \(1999\)](#) we use the fraction of zero returns days as a measure of liquidity. This paper shows the significant positive correlation of transaction cost and zero returns days (zero volume

days). Since high transaction cost usually means low liquidity, in our research, we use this as a measure of illiquidity for bond at month t and define it as $P_{t,zero}$ which is defined as

$$P_{t,zero} = \frac{N_{t,zero}}{N_{t,Total}}.$$

$N_{t,zero}$ is the number trading days in month t where no transaction for that bond and $N_{t,Total}$ is the total number of trading days in month t .

Average Bid-Ask Spread (BIDASK): In our research, for month t , day d , where there is at least one buy and one sell transaction, we use the difference between the average buy and the average sell price on each trading day to calculate the bid-ask spread for that day. The monthly bid-ask spread is calculated by averaging the daily average bid-ask spread.

$$BIDASK_{t,d} = \frac{P_{t,d,ask} - P_{t,d,bid}}{0.5(P_{t,d,ask} + P_{t,d,bid})}$$

$$BIDASK_t = \frac{1}{n} \sum_{d=1}^n BIDASK_{t,d}$$

where $BIDASK_t$ is the average bid ask spread for month t .

3.3.3.5 Momentum and short-term reversal

Short-term reversal (REV): We use the bond return in the previous month, if any, as the short-term reversal factor.

Short-term momentum (MOM6): Short-term momentum for month t is defined as the cumulative log return on the bond from month $t - 6$ to month $t - 1$. Following the momentum strategies by [Jostova et al. \(2013\)](#), we skip month t to avoid potential biases from bid-ask bounce and short-term price reversal.

Long-term momentum (MOM12): Similar to MOM6, long-term momentum for month t is defined as the cumulative log return on the bond from month $t - 13$ to $t - 1$. We also skip month t for the same reason discussed in constructing MOM6.

3.3.3.6 Distributional characteristics

[Bai et al. \(2016\)](#) suggested that the distributional characteristics of corporate bond predict the cross-sectional differences in future bond returns. Instead of a 60-month rolling window in the original article, we use a 36-month rolling window estimation to generate the monthly time series measure of volatility, skewness, and kurtosis for each bond defined below.

Volatility (VOL):

$$VOL_t = \frac{1}{n-1} \sum_{i=t-35}^t (R_i - \bar{R})^2$$

Skewness (SKEW):

$$SKEW_t = \frac{1}{n} \sum_{i=t-35}^t \left(\frac{R_i - \bar{R}}{\sigma} \right)^3$$

Kurtosis (KURT):

$$KURT_t = \frac{1}{n} \sum_{i=t-35}^t \left(\frac{R_i - \bar{R}}{\sigma} \right)^4 - 3$$

where $\bar{R}$, σ is the sample mean and sample standard deviation of bond returns, respectively, R_t is the log return for month t , and n is the actual number of nonzero observations in the rolling 36-month time period.

3.3.4 Stock characteristics

In the report by [Green et al. \(2017\)](#), the authors came up with 94 equity market characteristics and used them to predict the cross-sectional monthly stock returns. We leverage on their research and use the same 94 equity market characteristics to see if they have predictive power on corporate bond returns. The equity characteristics could be calculated using the SAS studio in the WRDS database.

3.3.5 Joining stock and bond characteristics

The primary key for equity characteristics is date and PERMNO, a unique permanent security identification number assigned by the CRSP to each security, while the primary key for our monthly bond returns are date and CUSIP. In order to join bond characteristics with equity characteristics, a linking table is needed to find the corresponding PERMNO for each CUSIP for a given month. Since TRACE provides company symbol for each CUSIP, we could look up the company symbol in the CRSP database to get the corresponding PERMNO. With the linking table, we then match the TRACE and CRSP by the company symbol as the bridge. However, because these two databases do not update this information at the same time when the trading symbol changes, we need to impose a conservative date range to reduce our error rate of matching. For example, the link start date is the maximum start date in TRACE and CRSP, and the link end date is the minimum of the end date in the two databases.

3.3.6 Dealing with missing data

Unlike the equity market, the corporate bond market is much more illiquid, and most of the bonds do not trade each day. Thus, it is impossible to calculate the daily returns. Table 3.2 shows that even if we focus only on a monthly frequency, we still have a lot of missing values. Since our forecast horizon is one month in the future, we filter out the records for each month's CUSIP combination if it does not trade in the following month. However, some of our features need to be calculated using the returns from prior months, and these features have the most missing values. For the models we perform comparative analysis on, a model like XGBoost allows for missing data in its input, while models like Ridge, Lasso, and linear regression do not allow for missing data. Therefore, we need to design a schema for handling missing data. The approach we decide to take is to fill the missing data in the corresponding fields with their same period median. The rationale behind this is that we believe that bond characteristics are not stationary over time, and we chose to use the median to prevent the impact from outliers.

Feature type	Feature name	# Rows with value	Pcnt. missing
Bond-level characteristics	RATE	1316322	2.536%
	TMT	1350573	0.000%
	AGE	1350503	0.005%
	DUR	1340974	0.710%
Downside risk characteristics	VAR5	1031640	23.615%
	VAR10	1031640	23.615%
	ES5	1031640	23.615%
	ES10	1031640	23.615%
Systematics risk characteristics	MKT	1059638	21.542%
	DEF	1059638	21.542%
	TERM	1059638	21.542%
Bond-level Liquidity characteristics	ILLIQ	295843	78.095%
	ROLL84	295844	78.095%
	LIQZERO	590987	56.241%
	BIDASK	1197742	11.316%
Bond momentum and reversal	REV	1219779	9.684%
	MOM6	1147077	15.068%
	MOM12	1029854	23.747%
Bond-level distributional characteristics	VOL	1081878	19.894%
	SKEW	1081878	19.894%
	KURT	1081878	19.894%

Table 3.2: Missing data analysis

3.3.7 Exploratory data analysis of our dataset

Our complete TRACE data consist of 68,368 bonds issued by a total of 3677 firms, yielding 1524292 bond-month return observations from July 1st 2002 to September 30th 2020. The average time-to-maturity of bond is 6.67 years and the distribution is displayed in Figure 3.3. The average rating score is 8.28(BBB+), and for the 28,935 bonds that have a rating, 79.6% of the bonds are issued as IG bonds, and 20.4% of the bonds are issued as HY bonds. If weighted by offering volume, 78.7% are issued as IG, 16.9% are issued as HY, and 4.4% are issued with no rating. Figure 3.4 displays the rating number weighted by offering par amount.

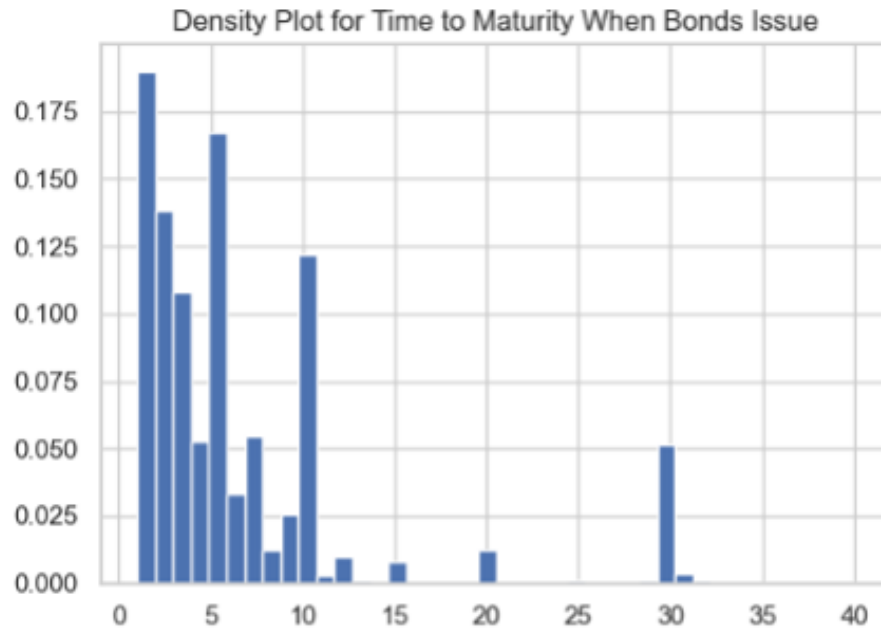


Figure 3.3: Density plot for time-to-maturity when bonds issue

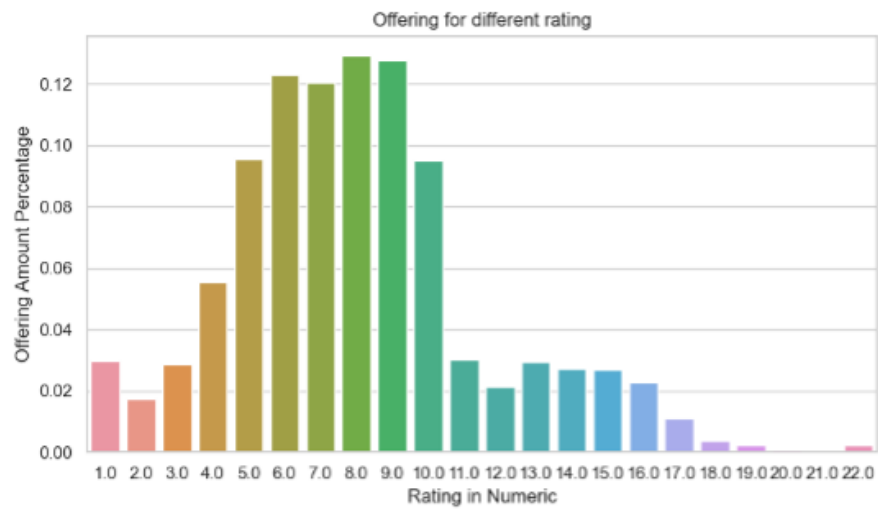


Figure 3.4: Rating of bonds

Table 3.3 displays the summary statistics for each variable.

	mean	std	min	25%	75%	max	Kurtosis	Skewness
NXT_EXCESS	0.00288	0.04614	(1.00110)	(0.00872)	0.01355	0.99997	104.68508	2.58718
RATE	8.68810	3.55984	1.00000	6.00000	10.00000	22.00000	0.59189	0.72193
TMT	9.79546	9.70233	1.08000	3.59000	11.21000	101.88000	16.02765	2.84762
AGE	5.05696	4.86804	(1.98768)	1.62628	6.85558	47.79466	3.61816	1.78792
DUR	6.24142	4.12510	0.07000	3.13000	8.05000	30.00000	0.33152	1.02994
VAR5	(0.06458)	0.07415	(0.98980)	(0.07201)	(0.02514)	0.00841	25.89069	(4.07993)
VAR10	(0.06420)	0.07354	(0.98980)	(0.07164)	(0.02503)	0.00843	25.73842	(4.06962)
ES5	(0.06473)	0.07440	(0.98980)	(0.07216)	(0.02518)	0.00841	25.95241	(4.08409)
ES10	(0.06454)	0.07409	(0.98980)	(0.07197)	(0.02513)	0.00841	25.87533	(4.07889)
MKT	1.00228	0.83810	(20.22391)	0.49321	1.36119	24.22125	29.56299	2.50112
DEF	0.09751	1.53419	(60.88349)	(0.63744)	0.47670	48.49724	41.23550	3.53029
TERM	(0.04519)	0.65620	(21.64807)	(0.17453)	0.23877	25.23744	63.71860	(4.22551)
ILLIQ	0.00019	0.01024	(0.03938)	0.00000	0.00007	2.71768	39,060.75769	183.84282
ROLL84	0.01284	0.02452	—	0.00370	0.01619	3.29708	4,537.47057	47.18063
LIQZERO	0.56232	0.32800	0.04000	0.25000	0.90000	1.00000	(1.42192)	(0.07983)
BIDASK	0.00654	0.01236	—	0.00210	0.00780	1.96700	4,992.40296	47.34155
REV	0.00629	0.04440	(0.98970)	(0.00435)	0.01640	1.00000	106.44180	2.66297
MOM6	0.03516	0.11209	(0.99518)	0.00301	0.05926	9.85721	423.51477	10.91338
MOM12	0.07083	0.17831	(0.99266)	0.01690	0.10217	25.00035	1,492.90198	21.36323
VOL	0.03006	0.03006	0.00087	0.01415	0.03440	0.75902	34.71166	4.60170
SKEW	0.02038	0.84123	(5.67686)	(0.42565)	0.44412	5.66671	3.43342	0.04999
KURT	1.39820	2.70149	(1.99907)	(0.20392)	1.95434	30.50327	11.95133	2.84022

Table 3.3: Summary statistics for each variable

3.3.8 Bond filter rules

Before model training, we perform the following bond filtering rules. (i) filter out perpetual bonds, convertible bonds, and preferred securities; (ii) filter out bonds that de-

nominated in foreign currency; (iii) filter out bonds with less than one year to maturity; (iv) only include zero-coupon bonds or semiannual coupon bonds; (v) filter out bond's observations after it trades below 20% of par value or trade above 200% of par value. The final dataset consists of 34,592 unique CUSIPs and 1,350,573 observations.

3.4 Predictability of bond returns

3.4.1 Out-of-sample predictability of bond returns using bond characteristics

In the table below, we present monthly out-of-sample R-squared (R^2_{oos}) values for the corporate bonds using bond characteristics in the testing period from 2017 to 2020 for all machine learning models and our OLS benchmark. In this dataset, all the models—including the benchmark OLS model—report R^2_{oos} values larger than zero. This indicates that all models deliver significant out-of-sample forecasting power for the expected returns on corporate bonds. In contrast to the study by [Gu et al. \(2020\)](#) and [Bali et al. \(2020\)](#), in our study, the benchmark OLS model has a positive R^2_{oos} instead of a negative one. We believe that this is due to the discrepancy between the testing periods. The testing period for the above papers is more than 20 years in length, and our testing period is from 2017 to 2020 and has less than 4 years of data. Since OLS does not have any hyperparameters, cross-validation is unnecessary. For the other models, model performance is calculated by setting the hyperparameters that give the lowest cross-validation error. Among all the models, the best is XGBoost, which has an R^2_{oos} of 5.60%, followed by the neural network, which has an R^2_{oos} of 5.06%. In this dataset, all other models have

R^2_{oos} higher than the OLS except for the decision tree, which records an R^2_{oos} of 2.97%. Another point that is worth mentioning is that we fit two XGBoost models: one using input data that contain missing values (XGBoost [with nan]) and another using input data that do not contain any missing values (XGBoost [w/o nan]). The model containing missing values outperform the other with a difference in R^2_{oos} of 0.30% in difference. Since the XGBoost implementation of boosting tree is able to handle missing data in tree construction by directly feeding raw data before missing value imputation, we can increase the performance of the model as imputation would introduce noise to the dataset. This is an interesting finding because in fixed income—especially in the world of corporate bonds—the data typically has a low signal-to-noise ratio and contains many missing values. A method like XGBoost, which can accept missing data and take advantage of that in tree construction, would be a superior choice when dealing with datasets with many missing values.

Model	Optimal hyperparameters
OLS	N/A
Ridge	Lambda = 8,
Lasso	Lambda = 0.001,
Decision tree	Tree depth = 2,
Random Forest	N estimators = 700, max depth = 5,
XGBoost (with nan)	n estimators = 100, max depth = 4, learning rate = 0.05, colsample bytree = 0.5, subsample = 0.5,
XGBoost (w/o nan)	n estimators = 300, max depth = 2, learning rate = 0.02, colsample bytree = 0.5, subsample = 0.8,
Neural network	Hidden layer size=7, number of neurons=32, activation = “relu”, Solver = “adam”, batch size = 10000, learning rate = “constant”, Learning rate init = 0,001, max iteration = 200.

Table 3.5: Optimal hyperparameters for bond features model

Model	Test R^2
OLS	4.09%
Ridge	4.38%
Lasso	4.83%
Decision tree	2.97%
Random forest	4.44%
XGBoost (with nan)	5.60%
XGBoost (w/o nan)	5.30%
Neural network	5.06%

Table 3.4: Out-of-sample R^2 for bonds features

We also display the optimal hyperparameters for cross-validation and the hyperparameter search space below for reference.

Model	Hyperparameters search space
OLS	N/A
Ridge	Lambda = [1e ⁻¹⁰ , 1e ⁻⁴ , 1e ⁻³ , 1e ⁻² , 1e ⁻¹ , 0.5, 0.75, 1, 2, 3, 4, 5, 7, 8, 10, 50, 100, 500, 1000]
Lasso	Lambda = [1e ⁻¹¹ , 5e ⁻¹¹ , 1e ⁻¹⁰ , 5e ⁻¹⁰ , 1e ⁻⁹ , 5e ⁻⁹ , 1e ⁻⁸ , 5e ⁻⁸ , 1e ⁻⁷ , 5e ⁻⁷ , 1e ⁻⁶ , 5e ⁻⁶ , 1e ⁻⁵ , 5e ⁻⁵ , 0.0001, 0.0005, 0.001, 0.005]
Decision tree	Tree_depth = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20]
Random forest	N_estimators = [50, 100, 200, 300, 500] Max_features = ["sqrt"] Max_depth = [2, 5, 10, 20]
XGBoost	n_estimators = [100, 300, 500, 700] max_depth = [2, 4, 8] learning_rate = [0.02, 0.05, 0.1] colsample_bytree = [0.5] subsample = [0.4, 0.8]
Neural network	nn_n_hidden layers = [1, 3, 5, 7, 10] nn_n_neurons = [16, 32, 64, 80] nn_learning_rate = ["constant"] nn_learning_rate_init = [0.01, 0.001] activation = "relu"

Table 3.6: Hyperparameters search space for bond features only model

3.4.2 Out-of-sample predictability of bond returns using bond and equity characteristics

Using the method described in the previous section, we join bond characteristics with equity characteristics by matching company identifiers in the equity market with

bond identifiers. By linking these two datasets, the number of features increases from 21 to 115; therefore, the models have more data to work with. However, we still face issues with missing data; as among the 33,498 bonds in our data set, only around 80% of them are from a company that has stock traded, which means that 80% of the record does not have any equity-related field. Using the same setup as the previous subsection, we run the models using the combined dataset and present R^2_{oos} for the corporate bonds using bond characteristics as well as the equity characteristics in the testing period from 2018 to 2020 for all machine learning models as well as our OLS benchmark in Table 3.7 below.

With equity data, our benchmark model's R^2_{oos} increases from 4.09% to 4.49%, indicating that equity data might enhance our prediction. Similarly, the other two linear models (Ridge and Lasso) display R^2_{oos} higher than the model with only bond characteristics. Ridge's R^2_{oos} increases from 4.38% to 4.69% and Lasso's R^2_{oos} increases from 4.83% to 5.11%. On average, the tree-based model also benefits from the extra data. Random forest's R^2_{oos} increases from 4.44% to 5.20%, and our best model XGBoost(w/o nan), record an R^2_{oos} of 6.70% and outperforms the best model when we only have bond data by more than 1%. In general, by introducing more data like equity features, better performance could be achieved by tree-based models. The only exception is the neural network. After adding equity features, the traditional feed-forward neural network seems to overfit the data. Advanced regularization techniques, such as dropout and early stopping, should be used to prevent overfitting and are left for future studies.

Model	Hyperparameters search space
OLS	N/A
Ridge	$\text{Lambda} = [1e^{-10}, 1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}, 0.5, 0.75, 1, 2, 3, 4, 5, 7, 8, 10, 50, 100, 500, 1000]$
Lasso	$\text{Lambda} = [1e^{-11}, 5e^{-11}, 1e^{-10}, 5e^{-10}, 1e^{-9}, 5e^{-9}, 1e^{-8}, 5e^{-8}, 1e^{-7}, 5e^{-7}, 1e^{-6}, 5e^{-6}, 1e^{-5}, 5e^{-5}, 0.0001, 0.0005, 0.001, 0.005]$
Decision tree	$\text{Tree_depth} = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20]$
Random forest	$\text{N_estimators} = [50, 100, 200, 300, 500]$ $\text{Max_features} = [\text{"sqrt"}]$ $\text{Max_depth} = [2, 5, 10, 20]$
XGBoost	$\text{n_estimators} = [100, 300, 500, 700]$ $\text{max_depth} = [2, 4, 8]$ $\text{learning_rate} = [0.02, 0.05, 0.1]$ $\text{colsample_bytree} = [0.5]$ $\text{subsample} = [0.4, 0.8]$
Neural network	$\text{nn_n_hidden layers} = [1, 3, 5, 7, 10]$ $\text{nn_n_neurons} = [16, 32, 64, 80]$ $\text{nn_learning_rate} = [\text{"constant"}]$ $\text{nn_learning_rate_init} = [0.01, 0.001]$ $\text{activation} = \text{"relu"}$

Table 3.8: Hyperparameter search space for bond and equity features model

Model	Test R^2
OLS	4.49%
Ridge	4.69%
Lasso	5.11%
Decision tree	3.08%
Random forest	5.20%
XGBoost (with nan)	6.19%
XGBoost (w/o nan)	6.70%
Neural network	3.46%

Table 3.7: Out-of-sample R^2 for bond and equity features

We also display the optimal hyperparameters for cross-validation and the hyperparameters search space below for reference.

Model	Optimal hyperparameters
OLS	N/A
Ridge	Lambda = 8,
Lasso	Lambda = 0.001,
Decision tree	Tree depth = 1,
Random forest	N estimators = 100, max depth = 5,
XGBoost (with nan)	n estimators = 100, max depth = 2, learning rate = 0.1, colsample bytree = 0.5, subsample = 0.5,
XGBoost (w/o nan)	n estimators = 300, max depth = 2, learning rate = 0.02, colsample bytree = 0.5, subsample = 0.5,
Neural network	Hidden layer size=1, number of neurons=64, activation = “relu”, Solver = “adam”, batch size = 500, learning rate = “constant”, Learning rate init = 0.01, max iteration = 200.

Table 3.9: Optimal hyperparameters for bond and equity features model

3.5 Feature Importance

Feature importance is a class of techniques that usually assigns a score to input features based on how useful they are when the model makes predictions. The main reason people perform such analyses and the same reason we have this section is to investigate which features the models rely on when making a prediction. Moreover, the feature importance result presented in this section is used in the next section on variable selection

to build parsimonious models. To calculate our models' feature importance score, we test three different approaches. The first approach is by using scikit-learn's package default for each model, the second approach is using reduction in R^2 as shown by [Kelly et al. \(2019\)](#) and [Gu et al. \(2020\)](#), while the third approach is by using SHAP introduced by [Lundberg and Lee \(2017\)](#). Later in this chapter, it will be shown that the SHAP method has the most realizable and consistent measure.

3.5.1 The naive approach

Figure 3.6 displays the feature importance score for package default for scikit-learn. This package does not provide feature importance for models like OLS, Ridge, Lasso, and neural networks. For linear models, feature importance in the above graph is proxy by the absolute value of the model's coefficients after standardizing the input feature matrix. For the tree-based models, the package default is mean decrease in impurity. The feature importance for the neural net is just not displayed here.

For linear models, the most important features are VAR10, VAR5, ES10, ES5, and VOL, which are from two categories. The first four measure downside risk characteristics, while VOL is classified as a distributional characteristic but related to downside risk. Since the decision tree has low depth, we only have three nonzero feature importance. The top three features for the random forest model are ROLL84, MOM12, and ILLIQ. ROLL84 and ILLIQ are liquidity measures, and MOM12 is momentum and short-term reversal. As for the XGBoost model, the top three most important features are ES5, VAR10, and VAR5. These features are all from downside risk characteristics, indicating

that downside risks are important features. Despite the simplicity of this approach, this method suffers from one big drawback: we could not use the same feature importance metric across models, and in some models like the neural network, the feature importance score is just not provided.

Feature Importance of Models based on scikit-learn default

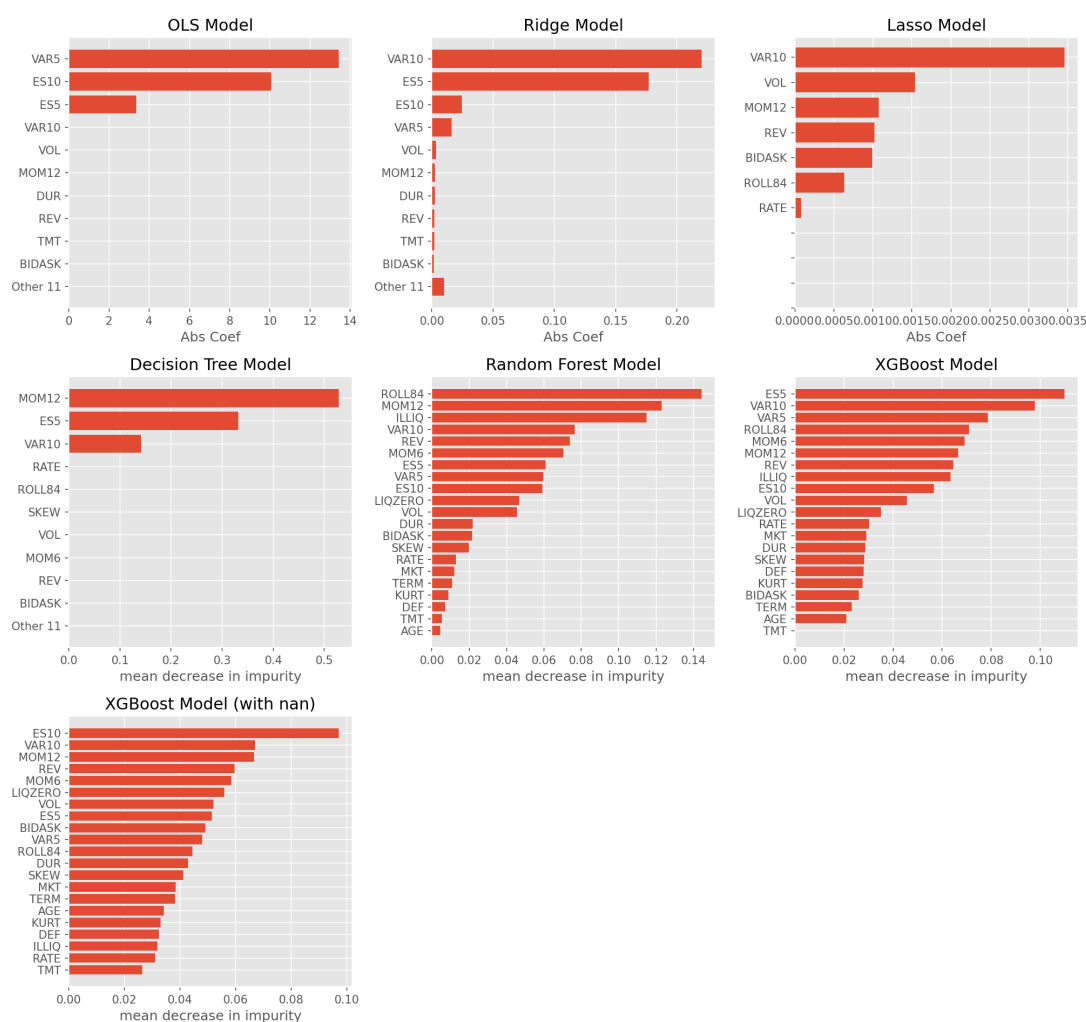


Figure 3.5: Feature importance of models from scikit-learn default

3.5.2 Feature importance using reduction in R-squared value

If we want a model agnostic method, another approach is using reduction in R^2 from setting the values of all observation's j -th predictor to zero while holding the remaining model estimates fixed. This is a simple yet popular approach and it is used in almost all relevant literature. When using this approach, the most important features for linear models are similar to those in the previous method—VAR10, VAR5, ES10, and ES5. It is evident that downside risk measures are the dominant features in all three linear models. For the decision tree, the dominant feature is ES5. Interestingly, in random forest and XGBoost, duration is the most dominant feature, and it accounts for 99% of the reduction in R^2 . Also, we observe the same pattern in the neural network. The explanation of this erroneous behavior is because zero in duration has a special meaning. Duration is the weighted average time to maturity of a bond, and a zero value means the bond is already mature. A mature bond should have zero return. In the tree splitting algorithm, if we change the feature DUR to zero, then in all the branches, when we split by the feature DUR, we will consistently go in one direction instead of having a choice. This behavior reveals the biggest drawback of this method as it failed to provide a realistic explanation of feature importance for the models. The actual contribution of one feature should be analyzed with other features to consider their interactions. When using this method, the other drawback is that we record a negative value in the reduction of R-squared in some of the models. For example, MOM12 in the random forest model and VAR10 in the decision tree. The model suggests that we could improve our prediction if we remove some of the features.

Feature Importance of Models based on Reduction of R-Squared

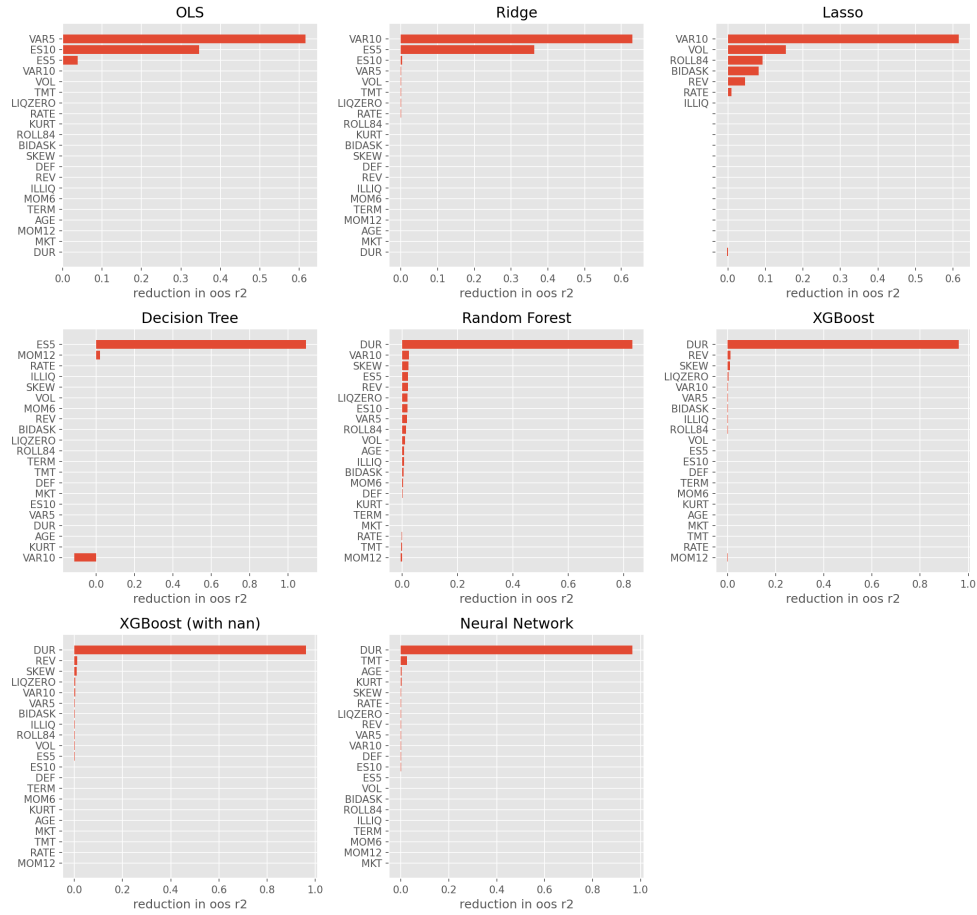


Figure 3.6: Feature importance of models based on reduction of R^2

3.5.3 Feature importance using SHAP

SHAP (SHapley Additive exPlanations) is a method based on Shapley value, a solution concept in cooperative game theory introduced by Lloyd Shapley in 1951. In SHAP, each feature value acts as a player in a coalition game, and Shapley values tell us how to distribute the “payout” fairly among features. There are three advantages of SHAP compared to other feature importance metrics. Firstly, SHAP could be model agnostic when

using Kernel SHAP. Secondly, SHAP provides feature importance explanation for each observation where most other methods usually do not. Thirdly, SHAP has consistency property, which says that if a model changes, such the marginal contribution of a feature changes, the Shapley value will also change in the same direction.

Figure 3.7 below displays the feature importance result for each machine learning model using SHAP. Each subplot displays the feature importance for a specific model. The x-axis is the average absolute Shapley value, and each feature is ranked by this value in a descending order. The first observation is that for linear models, the top features rankings are almost the same as using reduction in R-squared, where the top features are downside risk characteristics with illiquidity measures like ROLL84 and BIDASK. For the decision tree, the dominant feature is ES5, which is also the same as when we use reduction in R-squared. For random forest, the most important feature is VOL. Aside this, the top features for are mostly from downside risk measures and illiquidity measures. XGBoost (w/o nan)'s most important features are similar to those in random forest as it also relies heavily on downside risk and liquidity; however, the difference is that this method weights more on momentum and reversal. For XGBoost (with nan), the top 5 features are TMT, DUR, VOL, REV, and MOM12. They are mostly bond characteristic features, momentum, and reversal characteristics. It is quite interesting that XGBoost (w/o nan) weighs much heavier on liquidity measures than XGBoost (with nan). We believe this phenomenon is explained by how we calculate features from these groups. We need sufficient historical data to provide a good estimate when we calculate the features. For example, in the construction process for VOL and MOM12, if we have less than four monthly returns in the prior 36 months, we will mark it as missing. By doing this,

the liquidity information is actually embedded in these features. Furthermore, since the XGBoost model allows for input values to be missing and can use the sparsity-aware split finding algorithm [Chen and Guestrin \(2016\)](#) to split on missing values, it makes sense that this model does not need to rely heavily on liquidity features. The last model on the chart is the neural network, and the two dominant features are DUR and TMT for this model.

Feature Importance of Models based on SHAP

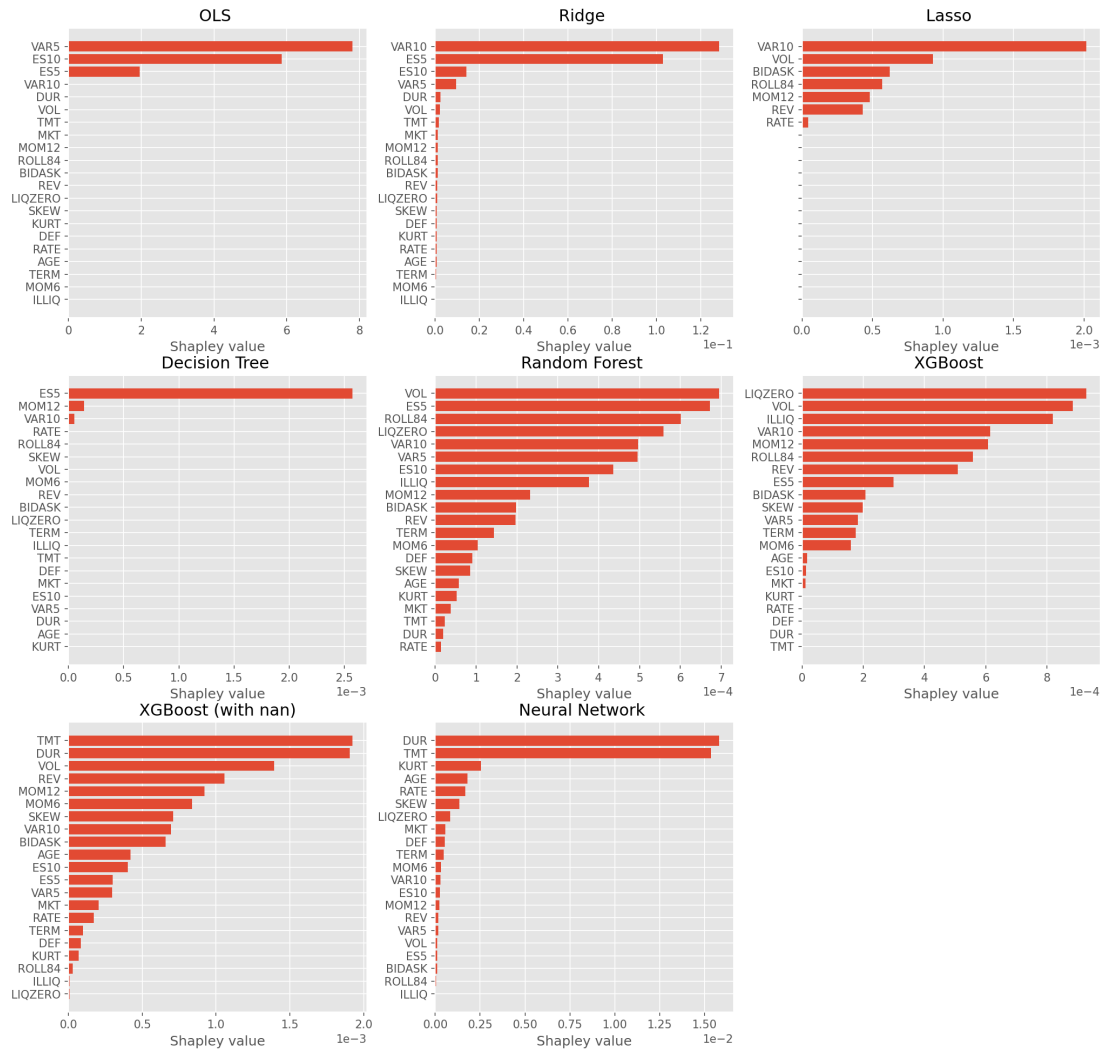


Figure 3.7: Feature importance based on SHAP(bond only)

Another advantage of SHAP is the ability to provide feature importance explanations for a single observation. For example, the figure below explains a single observation. (Noted that we multiply all values by 10,000 for better visualization). $E[f(x)]$ is the base value, which indicates that next month's excess return would be 84.859 basis points if we do not have any information from any features. $f(x)$ is our model prediction with all available information of all features. With this specific observation, we predict that the next month's excess return would be 95.279 basis points. Each row in the plot is a feature, and the number is the value for the feature we observe. For example, since we have the feature MOM equal to 0.1117, we would adjust our prediction upward by 22 basis points. Following the same logic, we adjust our prediction upward by 12.49 basis points from our observation of VOL and adjust downward by 8.33 basis points from our observation of ILLIQ.

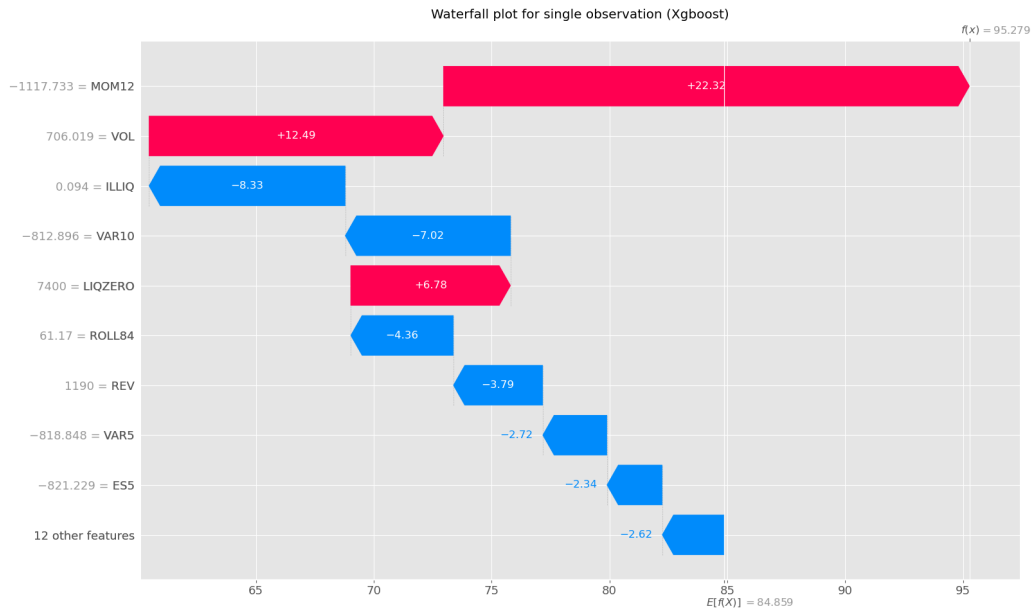


Figure 3.8: Feature importance for single observation(XGBoost)

To get a better understanding of the overall effect of top features. The plot below displays the SHAP feature importance across all models. The y-axis contains the feature names, which are ranked in descending order by the median value of the rank for each feature in different models. The x-axis shows the name of each model, and the value is the feature importance rank for each model. (with the most important feature ranked as one). From the plot below, we can see that VAR10 is the most important feature on average. The other top features come from downside risk measure, momentum and reversal, and illiquidity measure.

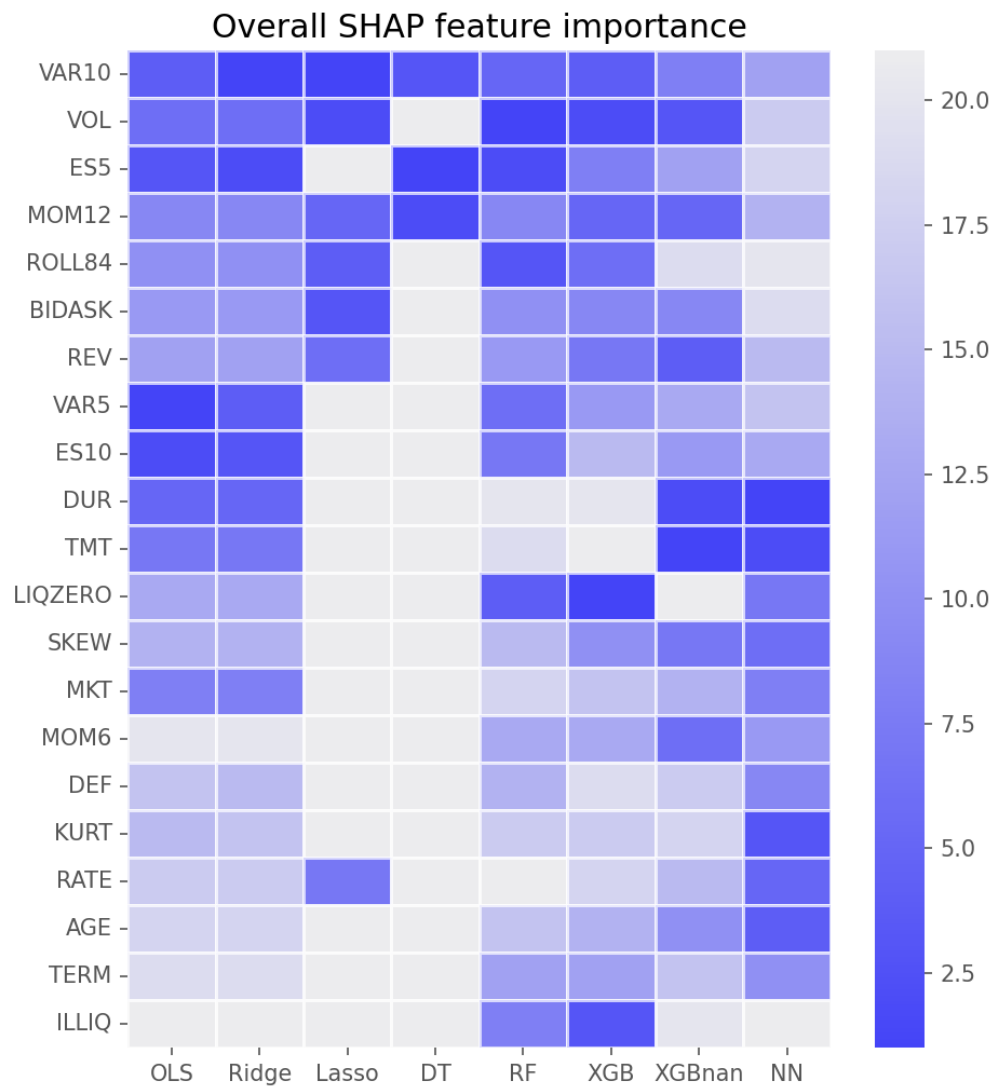


Figure 3.9: Overall SHAP feature importance(bond only)

3.5.4 Feature importance for bond and equity characteristics using SHAP

Adopting the SHAP method, we display below the top 15 most important features for each model calibrated using bond and equity characteristics. Despite the fact that we have more data coming from the equity market than those from the bond market (94 features versus 21 features), majority of the important features come from the bond market, especially for linear models. Moreover, the order of bond characteristics remains partially unchanged. For example, for Ridge regression, four of the top five features we calibrate using only bond features remain in the model we calibrate using bond and equity features. For lasso regression, two of the top three features remain. For the random forest, downside risk characteristics remain important as well. However, for XGBoost(with nan), only two of the top five features are the same, and the most important feature is now *rd_{mve}*, which comes from the equity market. For the neural network, *DUR* and *TMT* are still important but not as dominant. From observation, although the number of bond features is almost one-fifth those of equity, models lean towards using bond characteristics to predict individual bond returns.

However, two features in the equity market are relatively more important than others, and they are *baspread* and *retvol*. *baspread* is the monthly average of daily dollar bid-ask spread divided by the average of daily stock price, also called relative spread in the original paper ([Amihud and Mendelson, 1989](#)). Similar to the bond characteristic, *BIDASK*, *baspread* provides an estimate for the liquidity for individual bonds' parent company's stock instead of estimation of liquidity for the individual bond itself. Since the liquidity of individual bonds is generally high in noise, *baspread* serves as a good proxy,

which is reflected in the feature importance plot. Similarly, another bond characteristic `retvol` is defined as the standard deviation of daily returns on the stocks for individual bonds' parent company. This feature can act as the proxy for `VOL`, the standard deviation of the monthly returns for the bond itself.

Similar to the plot showing the overall importance of top features for models trained using bond characteristics, the plot below displays the SHAP feature importance across all models trained using bond and equity characteristics. The plot below shows that the top features are pretty stable after adding equity features. Two of the top five features remain, and they are `VAR10` and `VOL`. The most influential equity feature `baspread` that we discussed earlier is now ranked the second most important feature across all models.

Feature Importance of Models based on SHAP

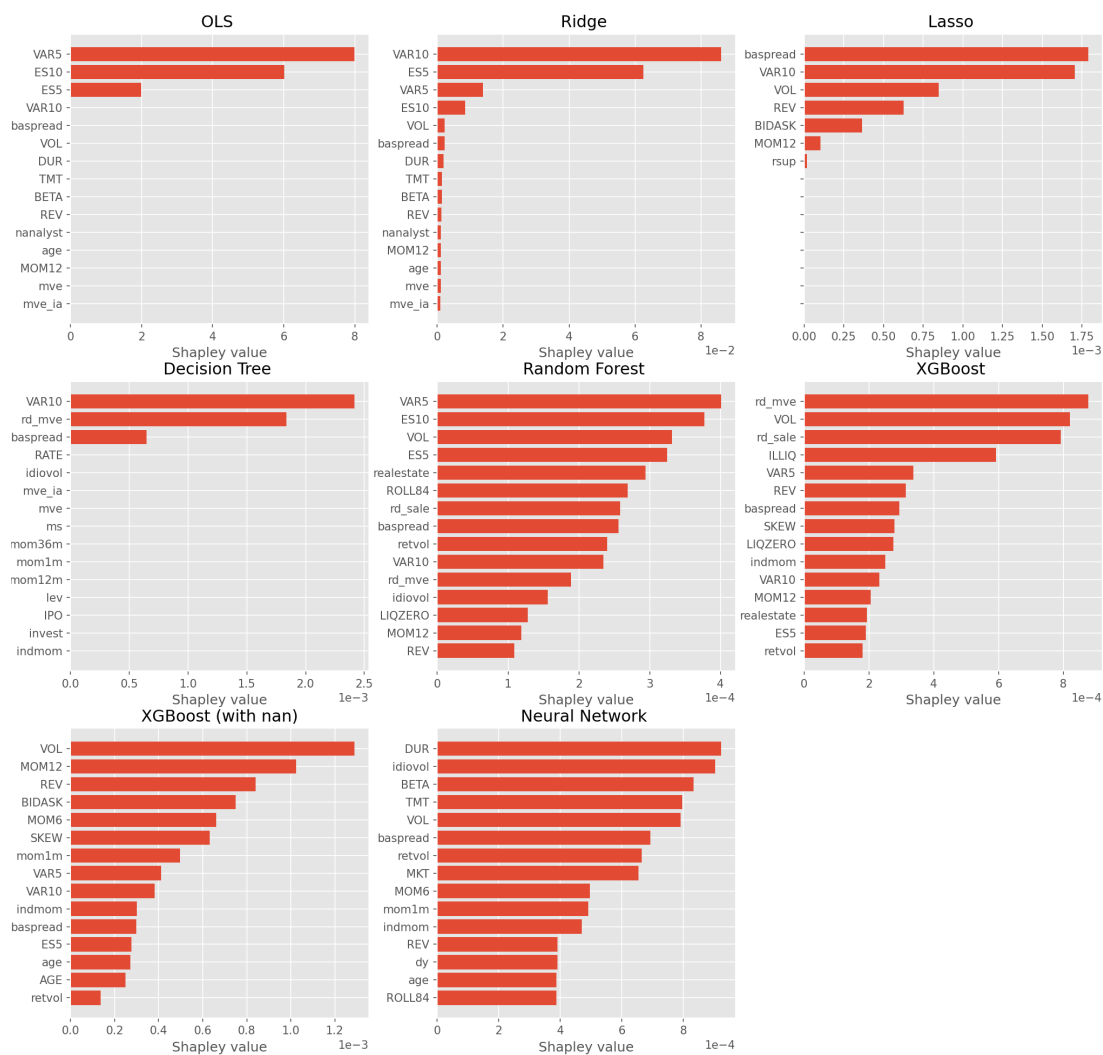


Figure 3.10: Feature importance of models based on SHAP(bond and equity)

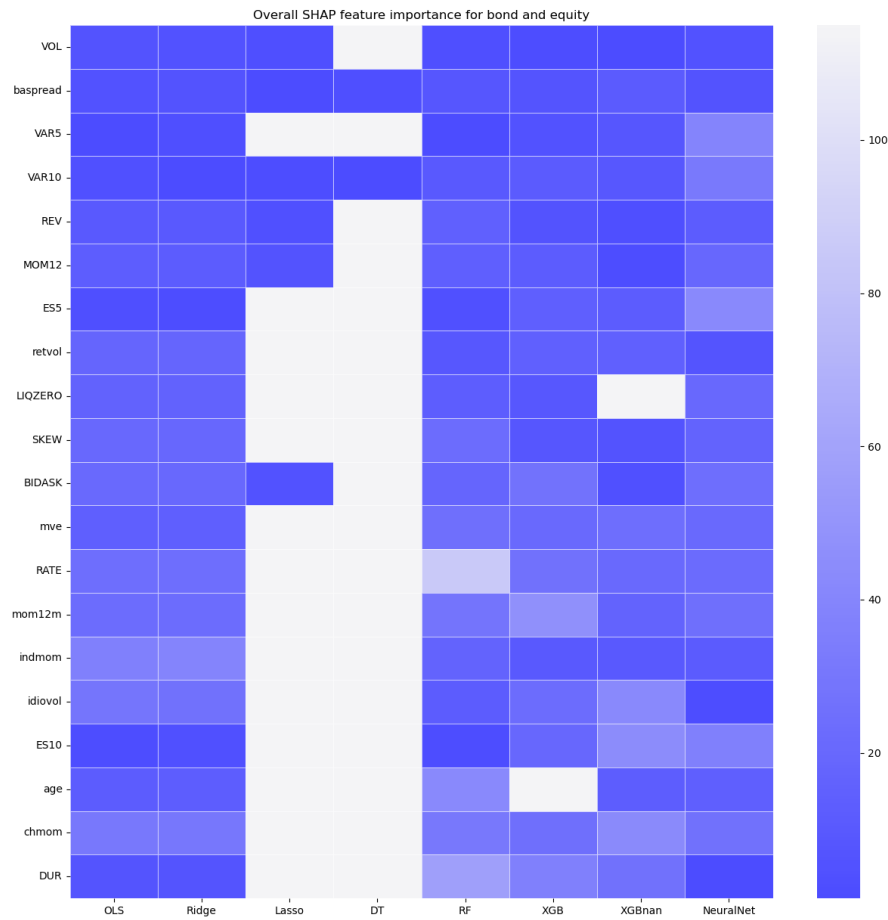


Figure 3.11: Overall SHAP Feature Importance for Bond and Equity

3.6 Machine Learning Portfolio

To investigate whether our prediction contributes to economic gains, we build a portfolio based on machine learning forecast using XGBoost with our bond characteristics. At the end of each month, we first calculate the 1-month-ahead out-of-sample

forecast using the model train by previous 3 months data. We then sort stocks into deciles based on our forecasts (decile one means top 10% outperforming bonds). We subsequently construct four equal-weighted portfolios based on the deciles. The ML long-only portfolio is constructed by longing the bonds in decile one. The ML short-only portfolio is constructed by shorting the bonds in decile ten, while the ML long-short is constructed by longing the decile one bonds while simultaneously shorting the decile ten bonds. Buy and hold is the benchmark portfolio that we buy all of the bonds available and hold till the end of the following month.

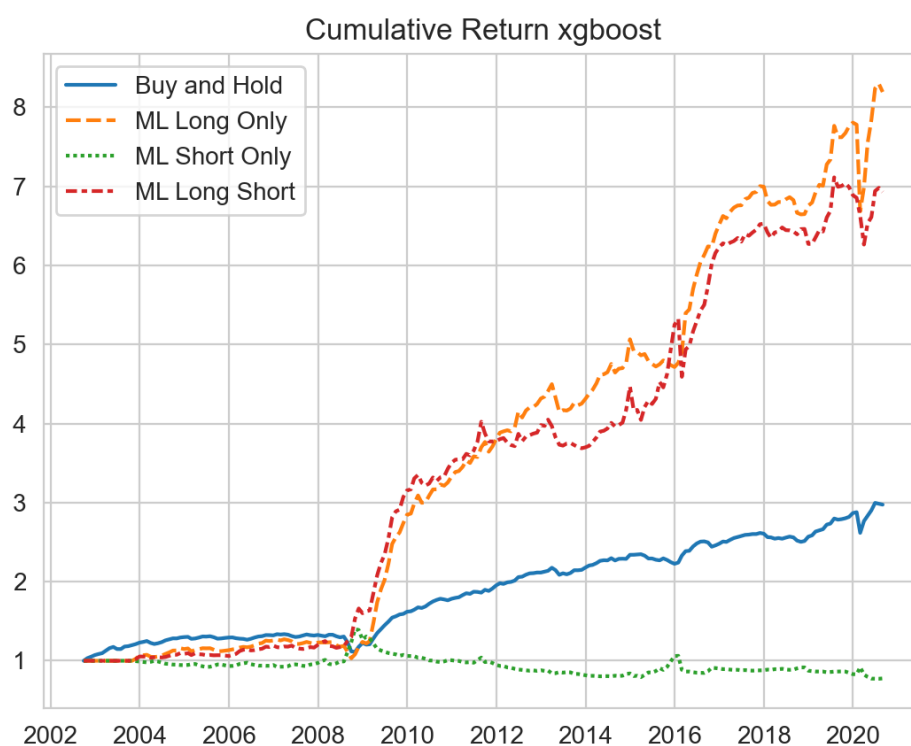


Figure 3.12: Cumulative return for XGBoost

Portfolio name	mean excess return(annualized)	Sharpe ratio (annualized)	max drawdown	max 1 month loss
Buy and Hold	5.21%	0.94	0.16	0.09
ML long-only	13.93%	1.31	0.19	0.12
ML short-only	-0.76%	-0.03	0.41	0.16
ML long-short	13.58%	1.27	0.14	0.14

Table 3.10: Key performance metric

Figure 3.12 displays the cumulative return of the four strategies. From the perspective of excess returns, we can tell from the graph that the best performing portfolio is the ML long-only portfolio, which has 1035% excess return over the period, and the worst portfolio is the ML short-only portfolio, which ends up worse than our benchmark portfolio. If we focus on risk-adjusted returns, the ML long-only portfolio is the winner, and it displays a Sharpe ratio of 1.31, around 40% increase in the Sharpe ratio compared to our benchmark portfolio.

Portfolio name	Average precision
Random Pick	10.00%
ML long-only	20.80%
ML short-only	14.13%
ML long-short	17.47%

Table 3.11: Average precision of our recommendation

In machine learning, a recommender system is a class of information-filtering systems that seek to predict the “rating” and “preference” that a user would give to an item. When we rebalance our portfolio at each month-end, we essentially try to predict the ac-

tual first decile and tenth decile portfolio using our model. We can treat the next month's portfolio as our recommendation in this context. For example, if we include a bond, A, in the next month's first decile portfolio and its realized returns rank as top 10%, we will count this bond as true positive. In contrast, if we include a bond, B, in the next month's first decile portfolio and its actual next month's return is not ranked as top 10%, we will count this bond as false positive.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Given the definition of precision, we calculate the average precision for each month for our portfolio, and the result is displayed in Table 3.11. Since we group our bonds into deciles, a portfolio consisting of random pick bonds theoretically has a precision of 10%. Our ML long-only (first decile) prediction has an average precision of 20.80%, much better than a random guess, and our ML short-only(tenth decile) prediction is the lowest. This could also explain why the ML short-only portfolio even performs worse than our benchmark. It is worth noting that in our scenario, the recall is the same as the precision because our false negative (the number of bonds that are actually in that decile but we fail to identify) is the same as our false positive (the number of bonds that we identify but are not in the decile).

To understand the performance of different machine learning trading strategies, we also backtest this strategy using different machine learning models.

Model	Strategy	Cum Return	Annual Excess Return	Sharpe Ratio	Max Drawdown	Max IM Loss	Avg Precision
Benchmark	Buy and Hold	172%	3.31%	0.525	0.385	0.093	N/A
OLS	ML long-only	759%	12.75%	1.27	0.18	−0.13	20.10%
	ML short-only	−45%	−3.30%	−0.35	0.52	−0.17	13.42%
	ML long-short	410%	9.52%	1.02	0.13	−0.13	16.74%
Ridge	ML long-only	796%	13.02%	1.23	0.20	−0.16	20.90%
	ML short-only	−32%	−2.12%	−0.19	0.46	−0.17	14.09%
	ML long-short	564%	11.14%	1.09	0.15	−0.14	17.48%
Lasso	ML long-only	811%	13.12%	1.23	0.20	−0.15	21.03%
	ML short-only	−30%	−1.93%	−0.17	0.45	−0.17	14.02%
	ML long-short	596%	11.44%	1.09	0.15	−0.14	17.51%
Decision tree	ML long-only	279%	7.71%	1.01	0.19	−0.12	12.69%
	ML short-only	−53%	−4.15%	−0.51	0.59	−0.15	11.34%
	ML long-short	92%	3.72%	0.63	0.12	−0.12	12.01%
Random forest	ML long-only	759%	12.75%	1.23	0.19	−0.12	18.46%
	ML short-only	−19%	−1.17%	−0.08	0.42	−0.17	13.23%
	ML long-short	650%	11.90%	1.11	0.15	−0.15	15.83%
XGBoost (w/o nan)	ML long-only	719%	12.46%	1.19	0.19	−0.14	18.61%
	ML short-only	−22%	−1.40%	−0.10	0.45	−0.17	13.51%
	ML long-short	593%	11.41%	1.11	0.14	−0.14	16.05%
XGBoost (with nan)	ML long-only	936%	13.93%	1.31	0.17	−0.12	20.84%
	ML short-only	−13%	−0.77%	−0.03	0.41	−0.16	14.13%
	ML long-short	879%	13.58%	1.27	0.14	−0.14	17.47%
Neural network	ML long-only	516%	10.67%	1.21	0.17	−0.09	16.23%
	ML short-only	−54%	−4.25%	−0.54	0.59	−0.13	14.46%
	ML long-short	207%	6.46%	0.92	0.10	−0.10	15.34%
Model average	ML long-only	697%	12.05%	1.21	0.18	−0.13	18.61%
	ML short-only	−34%	−2.39%	−0.25	0.49	−0.16	13.52%
	ML long-short	499%	9.90%	1.03	0.14	−0.13	16.05%

Table 3.12: Performances of machine learning portfolios

The chart above summarizes the return of all three strategies using different models.

Of the three strategies, we can see that the ML long-only strategies have the best average performance in all metrics. ML long-only Strategies have an average cumulative return

of 797%, the average annualized excess return of 12.05%, a Sharpe ratio of 1.21, and a maximum drawdown of 0.18. In contrast, ML short-only strategies are the least profitable strategies. The average cumulative return of -34% is less than 148% for the benchmark portfolio. The average annualized excess return and Sharpe ratio is lower than 0, also worse than the benchmark portfolio. In conclusion, the overall ranking of these three strategies is ML long-only, ML long-short, and ML short-only from the best to the worst.

For the long-only strategy, the best model is XGBoost(with nan). This model records an average excess return of 13.93% and Sharpe ratio of 1.31, followed by the Ridge regression model, which records an average excess return of 13.02% and Sharpe ratio of 1.23.

For long-short strategy, the two best models are XGBoost(with nan) and random forest, with a Sharpe ratio of 1.27 and 1.11, respectively.

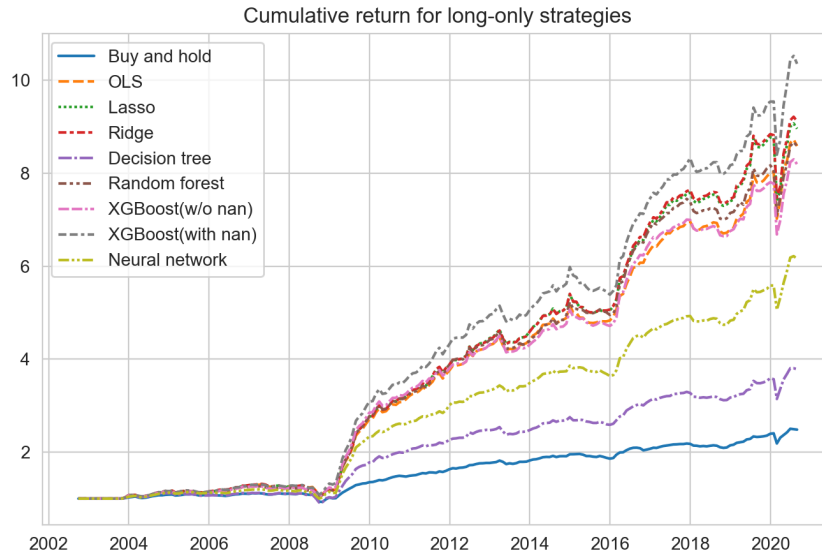


Figure 3.13: Cumulative return for ML long only strategy

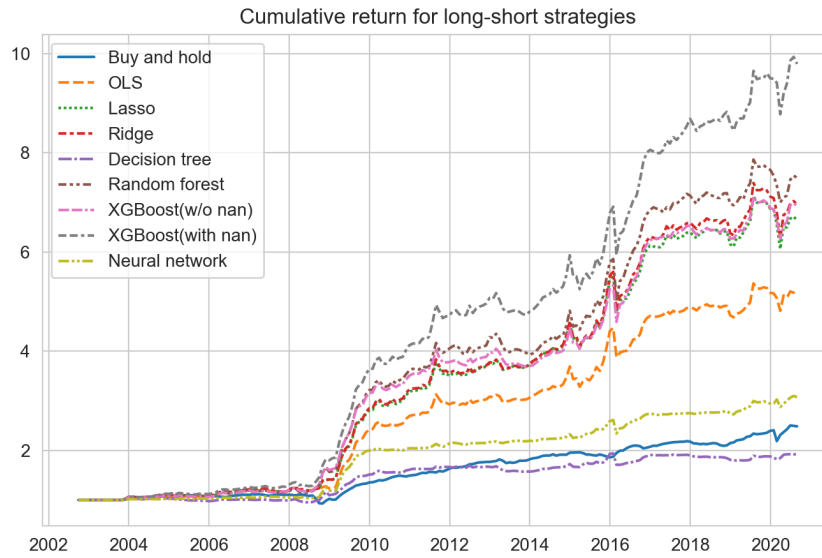


Figure 3.14: Cumulative return for ML long short strategy

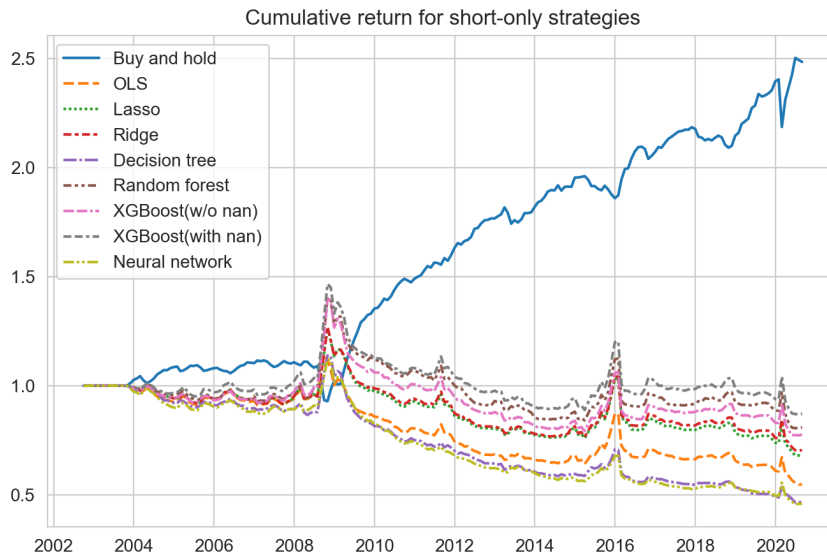


Figure 3.15: Cumulative return for ML short only strategy

The strategies from Table 3.12 are plotted in Figure 3.13, 3.14, and 3.15. These

three figures display the cumulative returns for ML long-only, ML long-short, and ML short-only strategies for machine learning models. We can confirm from the plot that no matter what model we choose, the ML short-only strategy fails to produce better cumulative returns than benchmark portfolios. In terms of strategies, the ML long-only outperforms ML long-short strategy in general. Moreover, it is worth mentioning that in terms of cumulative return, XGBoost(with nan) is the best model for both the ML long-short and ML long-only portfolios.

3.7 Parsimonious Model

The performance of machine learning models usually improves as the number of features increases. In the previous chapter, the best model we trained using bond and equity characteristics outperformed the best model we trained using bond characteristics. Theoretically, we prefer models that contain as many features as possible because they provide more information. However, a model with fewer features has its advantages in practice. One of the most significant advantages of a smaller model is its cost effectiveness and maintainability. Up-to-date financial data is generally hard to access without additional cost. If data is relatively expensive to acquire and only slightly improves the existing model, companies are most likely to give up on such data. Moreover, less data means the data pipeline is easier to maintain. From the operational point of view, a model with few features means less code to be written and less workforce is required for maintaining the data pipeline. Therefore, to be pragmatic, it is also crucial to investigate the performance of parsimonious models with fewer features.

A popular way of building parsimonious models is to start from a complex one and trim it down to one with fewer features or parameters. This paper uses the SHAP method presented above to perform feature selection. We leverage on SHAP to select the top 15 features, starting with the models with 115 bond and equity characteristics. With limited features, we use the cross-validation technique described previously to settle on hyperparameters and the result of the calibrated model is described below.

Model	Test R^2
OLS	4.16%
Ridge	4.36%
Lasso	5.11%
Decision tree	3.08%
Random forest	5.05%
XGBoost (with nan)	6.00%
XGBoost (w/o nan)	6.52%
Neural network	4.36%

Table 3.13: Out-of-sample R^2 for parsimonious model

Models that trained using their top 15 features display promising results. Compared to the best models with all available bond and equity features, the best parsimonious machine learning model is the XGBoost (w/o nan). The R^2_{oos} , only decreases from 6.70% to 6.52%. The performance of the parsimonious model is better than the model using only bond characteristic but worse than those using both bond and equity characteristics.

The table below presents the optimum hyperparameters and the hyperparameter search space for parsimonious models.

Model	Optimal hyperparameters
OLS	N/A
Ridge	Lambda = 8,
Lasso	Lambda = 0.001,
Decision tree	Tree depth = 2,
Random forest	N estimators = 200, max depth = 5,
XGBoost (with nan)	n estimators = 300, max depth = 2, learning rate = 0.02, colsample bytree = 0.5, subsample = 0.8,
XGBoost (no nan)	n estimators = 100, max depth = 2, learning rate = 0.05, colsample bytree = 0.5, subsample = 0.5,
Neural network	N/A

Table 3.14: Optimal hyperparameters for parsimonious model

Model	Hyperparameters search space
OLS	N/A
Ridge	Lambda = [1e ⁻¹⁰ , 1e ⁻⁴ , 1e ⁻³ , 1e ⁻² , 1e ⁻¹ , 0.5, 0.75, 1, 2, 3, 4, 5, 7, 8, 10, 50, 100, 500, 1000]
Lasso	Lambda = [1e ⁻¹¹ , 5e ⁻¹¹ , 1e ⁻¹⁰ , 5e ⁻¹⁰ , 1e ⁻⁹ , 5e ⁻⁹ , 1e ⁻⁸ , 5e ⁻⁸ , 1e ⁻⁷ , 5e ⁻⁷ , 1e ⁻⁶ , 5e ⁻⁶ , 1e ⁻⁵ , 5e ⁻⁵ , 0.0001, 0.0005, 0.001, 0.005]
Decision tree	Tree_depth = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20]
Random Forest	N_estimators = [50, 100, 200, 300, 500] Max_features = ["sqrt"] Max_depth = [2, 5, 10, 20]
XGBoost	n_estimators = [100, 300, 500, 700] max_depth = [2, 4, 8] learning_rate = [0.02, 0.05, 0.1] colsample_bytree = [0.5] subsample = [0.4, 0.8]
Neural network	nn_n_hidden layers = [1, 3, 5, 7, 10] nn_n_neurons = [16, 32, 64, 80] nn_learning_rate = ["constant"] nn_learning_rate_init = [0.01, 0.001] activation = "relu"

Table 3.15: Hyperparameters search space for parsimonious model

3.8 Conclusion and future research

Using machine learning methods, we comprehensively study the predictability of corporate bonds using 21 bond characteristics. Our findings suggest that machine learning methods can help improve our estimate of future bond returns. Ensemble tree-based mod-

els such as XGBoost is the best performing model. Both equity and corporate bonds are claims on companies. Thus, it is crucial to investigate whether including equity characteristics would have an improvement on our model. The models' performance increases after increasing the predictor to include 94 equity characteristics. The best performing model is still the XGBoost model after we include equity characteristics. We discuss methods for calculating feature importance and investigating the top features that the models rely on when predicting. We discuss the most popular way of calculating the feature importance called reduction of R^2 . The most significant drawback of this method is that it fails to consider the interaction between features and to address this problem, we propose using a new method called SHAP, which is based on Shapley value. The advantage of SHAP is that this method is model agnostic, consistent, and can provide feature importance explanation for each observation. To investigate whether our prediction contributes to economic gains, we build long-short, long-only, and short-only trading strategies for each model based on its prediction. The result shows that the long-only strategy using the XGBoost model's prediction is the most profitable strategy. In general, the performance of machine learning models usually improves as the number of features increases. However, in practice getting more data is costly, and maintaining a dataset with a large number of features could be challenging. We discuss a method of calculating parsimonious models by first training the model using the complete set of features and then retraining it using its most important features. Thus, we end up with a high-performance model with a few features.

Some ideas that worth exploring are left for future research. When implementing neural network, due to the low signal to noise ratio in corporate bond data, neural network

overfits the data easily and this limits its performance. To address this, it is worth exploring advance regularization techniques such as dropout and early stopping or exploring other types of networks. Due to the liquidity of corporate bonds, we conduct this research on monthly data. However, it is interesting to know the predictability of corporate bonds in a shorter time horizon.

Bibliography

- (2021). Bond returns by wrds. <https://wrds-www.wharton.upenn.edu/pages/get-data/wrds-bond-returns/>.
- (2021). Trade reporting and compliance engine (trace). <https://www.finra.org/filing-reporting/trace>.
- (2022). H.15 selected interest rates. <https://www.federalreserve.gov/releases/h15/>.
- Amihud, Y. and Mendelson, H. (1989). The effects of beta, bid-ask spread, residual risk, and size on stock returns. *The Journal of Finance*, 44(2):479–486.
- Anderson, T. W. and Darling, D. A. (1952). Asymptotic theory of certain” goodness of fit” criteria based on stochastic processes. *The annals of mathematical statistics*, pages 193–212.
- Andrade, S., Di Pietro, V., and Seasholes, M. (2005). Understanding the profitability of pairs trading. *Unpublished working paper, UC Berkeley, Northwestern University*.
- Archakov, I. and Hansen, P. R. (2021). A new parametrization of correlation matrices. *Econometrica*, 89(4):1699–1715.
- B Sabino da Silva, F., Ziegelman, F., and Caldeira, J. (2017). Mixed copula pairs trading strategy on the s&p 500. *Flávio and Caldeira, João, Mixed Copula Pairs Trading Strategy on the S&P, 500*.
- Bai, J., Bali, T. G., and Wen, Q. (2016). Do the distributional characteristics of corporate bonds predict their future returns?
- Bali, T. G., Goyal, A., Huang, D., Jiang, F., and Wen, Q. (2020). Different strokes: Return predictability across stocks and bonds with machine learning and big data. *Georgetown McDonough School of Business Research Paper*, (3686164):20–110.
- Bao, J., Pan, J., and Wang, J. (2011). The illiquidity of corporate bonds. *The Journal of Finance*, 66(3):911–946.
- Bernardo, A. E. and Ledoit, O. (2000). Gain, loss, and asset pricing. *Journal of political economy*, 108(1):144–172.

- Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. In *World Scientific Reference on Contingent Claims Analysis in Corporate Finance: Volume 1: Foundations of CCA and Equity Valuation*, pages 3–21. World Scientific.
- Bloomberg (2021). Bloomberg Barclays methodology.
- Bowen, D. A. and Hutchinson, M. C. (2016). Pairs trading in the uk equity market: Risk and return. *The European Journal of Finance*, 22(14):1363–1387.
- Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Cart. Classification and Regression Trees*, Wadsworth and Brooks/Cole, Monterey, CA.
- Buchmann, B., Lu, K. W., and Madan, D. B. (2019). Weak subordination of multivariate lévy processes and variance generalised gamma convolutions. *Bernoulli*, 25(1):742–770.
- Butaru, F., Chen, Q., Clark, B., Das, S., Lo, A. W., and Siddique, A. (2016). Risk and risk management in the credit card industry. *Journal of Banking & Finance*, 72:218–239.
- Caldeira, J. and Moura, G. V. (2013). Selection of a portfolio of pairs based on cointegration: A statistical arbitrage strategy. *Available at SSRN 2196391*.
- Campbell, J. Y. and Thompson, S. B. (2008). Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies*, 21(4):1509–1531.
- Carrasco, M., Chernov, M., Florens, J.-P., and Ghysels, E. (2007). Efficient estimation of general dynamic models with a continuum of moment conditions. *Journal of econometrics*, 140(2):529–573.
- Carrasco, M. and Florens, J.-P. (2000). Generalization of gmm to a continuum of moment conditions. *Econometric Theory*, 16(6):797–834.
- Carrasco, M. and Katchoni, R. (2017). Efficient estimation using the characteristic function. *Econometric Theory*, 33(2):479–526.
- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.
- Cici, G., Gibson, S., and Moussawi, R. (2017). Explaining and benchmarking corporate bond returns. *Available at SSRN 2995626*.
- Cummins, M. and Bucca, A. (2012). Quantitative spread trading on crude oil and refined products markets. *Quantitative Finance*, 12(12):1857–1875.
- Dhaene, J., Kukush, A., and Linders, D. (2014). The multivariate black & scholes market: conditions for completeness and no-arbitrage. *Theory of Probability and Mathematical Statistics*, 88:85–98.

- Dick-Nielsen, J. (2009). Liquidity biases in trace. *The Journal of Fixed Income*, 19(2):43–55.
- Dick-Nielsen, J. (2014). How to clean enhanced trace data. *Available at SSRN 2337908*.
- Diebold, F. X. and Shin, M. (2019). Machine learning for regularized survey forecast combination: Partially-egalitarian lasso and its derivatives. *International Journal of Forecasting*, 35(4):1679–1691.
- Do, B. and Faff, R. (2010). Does simple pairs trading still work? *Financial Analysts Journal*, 66(4):83–95.
- Do, B. and Faff, R. (2012). Are pairs trading profits robust to trading costs? *Journal of Financial Research*, 35(2):261–287.
- Drechsler, Q. (2017a). <https://wrds-www.wharton.upenn.edu/pages/support/manuals-and-overviews/wrds-bond-return/cleaning-trace-data/wrds-clean-trace-enhanced-file/>.
- Drechsler, Q. (2017b). <https://wrds-www.wharton.upenn.edu/pages/support/manuals-and-overviews/wrds-bond-return/cleaning-trace-data/wrds-clean-standard-trace-file/>.
- Elliott, R. J., Van Der Hoek*, J., and Malcolm, W. P. (2005). Pairs trading. *Quantitative Finance*, 5(3):271–276.
- Engle, R. F. and Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276.
- Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1):3–56.
- Fama, E. F. and French, K. R. (2015). A five-factor asset pricing model. *Journal of financial economics*, 116(1):1–22.
- Ferreira, L. (2008). New tools for spread trading. *Futures*, 37(12):38–41.
- Feuerverger, A. and McDunnough, P. (1981). On the efficiency of empirical characteristic function procedures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 20–27.
- French, K. R. (2022). <http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/>.
- Gatev, E., Goetzmann, W. N., and Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies*, 19(3):797–827.
- Goyal, A. (2012). Empirical cross-sectional asset pricing: a survey. *Financial Markets and Portfolio Management*, 26(1):3–38.
- Green, J., Hand, J. R., and Zhang, X. F. (2017). The characteristics that provide independent information about average us monthly stock returns. *The Review of Financial Studies*, 30(12):4389–4436.

- Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273.
- Hansen, L. P. (1982). Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the econometric society*, pages 1029–1054.
- He, X., Feng, G., Wang, J., and Wu, C. (2021). Predicting individual corporate bond returns. *Available at SSRN*.
- Huang, W., Liu, Q., Rhee, S. G., and Wu, F. (2012). Extreme downside risk and expected stock returns. *Journal of Banking & Finance*, 36(5):1492–1502.
- Huck, N. and Afawubo, K. (2015). Pairs trading and selection methods: is cointegration superior? *Applied Economics*, 47(6):599–613.
- Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
- Jacobs, H. and Weber, M. (2015). On the determinants of pairs trading profitability. *Journal of Financial Markets*, 23:75–97.
- Jostova, G., Nikolova, S., Philipov, A., and Stahel, C. W. (2013). Momentum in corporate bond returns. *The Review of Financial Studies*, 26(7):1649–1693.
- Jurek, J. W. and Yang, H. (2007). Dynamic portfolio selection in arbitrage. In *EFA 2006 Meetings Paper*.
- Justel, A., Peña, D., and Zamar, R. (1997). A multivariate kolmogorov-smirnov test of goodness of fit. *Statistics & Probability Letters*, 35(3):251–259.
- Kaplan, P. D. and Knowles, J. A. (2004). Kappa: A generalized downside risk-adjusted performance measure. *Journal of Performance Measurement*, 8:42–54.
- Kelly, B. T., Pruitt, S., and Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524.
- Khandani, A. E., Kim, A. J., and Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787.
- Kim, J.-M., Kim, D. H., and Jung, H. (2021). Applications of machine learning for corporate bond yield spread forecasting. *The North American Journal of Economics and Finance*, 58:101540.
- Kotchoni, R., Carrasco, M., et al. (2019). The continuum-gmm estimation: Theory and application. *International Financial Markets [Book], Volume 1*.
- Krauss, C. (2017). Statistical arbitrage pairs trading strategies: Review and outlook. *Journal of Economic Surveys*, 31(2):513–545.

- Küchler, U. and Tappe, S. (2008). Bilateral gamma distributions and processes in financial mathematics. *Stochastic Processes and their Applications*, 118(2):261–283.
- Lesmond, D. A., Ogden, J. P., and Trzcinka, C. A. (1999). A new estimate of transaction costs. *The review of financial studies*, 12(5):1113–1141.
- Liew, R. Q. and Wu, Y. (2013). Pairs trading: A copula approach. *Journal of Derivatives & Hedge Funds*, 19(1):12–30.
- Lin, H., Wang, J., and Wu, C. (2011). Liquidity risk and expected corporate bond returns. *Journal of Financial Economics*, 99(3):628–650.
- Lin, Y.-X., McCRAE, M., and Gulati, C. (2006). Loss protection in pairs trading through minimum profit bounds: A cointegration approach. *Advances in Decision Sciences*, 2006.
- Liu, J. and Timmermann, A. (2013). Optimal convergence trade strategies. *The Review of Financial Studies*, 26(4):1048–1086.
- Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- MacKenzie, D. and Spears, T. (2014). ‘the formula that killed wall street’: The gaussian copula and modelling practices in investment banking. *Social Studies of Science*, 44(3):393–417.
- Madan, D. and Seneta, E. (1987). Simulation of estimates using the empirical characteristic function. *International Statistical Review/Revue Internationale de Statistique*, pages 153–161.
- Madan, D. B. (2015). Estimating parametric models of probability distributions. *Methodology and Computing in Applied Probability*, 17(3):823–831.
- Madan, D. B. (2020). Multivariate distributions for financial returns. *International Journal of Theoretical and Applied Finance*, 23(06):2050041.
- Madan, D. B., Carr, P. P., and Chang, E. C. (1998). The variance gamma process and option pricing. *Review of Finance*, 2(1):79–105.
- Madan, D. B. and Schoutens, W. (2022). *Nonlinear Valuation and Non-Gaussian Risks in Finance*. Cambridge University Press.
- Madan, D. B., Schoutens, W., and Wang, K. (2017). Measuring and monitoring the efficiency of markets. *International Journal of Theoretical and Applied Finance*, 20(08):1750051.
- Madan, D. B. and Seneta, E. (1990). The variance gamma (vg) model for share market returns. *Journal of business*, pages 511–524.

- Madan, D. B. and Wang, K. (2017). Asymmetries in financial returns. *International Journal of Financial Engineering*, 4(04):1750045.
- Malevergne, Y. and Sornette, D. (2003). Testing the gaussian copula hypothesis for financial assets dependences. *Quantitative finance*, 3(4):231.
- Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of economics and management science*, pages 141–183.
- Miao, G. J. (2014). High frequency and dynamic pairs trading based on statistical arbitrage using a two-stage correlation and cointegration approach. *International Journal of Economics and Finance*, 6(3):96–110.
- Nelsen, R. B. (2007). *An introduction to copulas*. Springer Science & Business Media.
- Niederhoffer, V. and Osborne, M. F. M. (1966). Market making and reversal on the stock exchange. *Journal of the American Statistical Association*, 61(316):897–916.
- Perlin, M. S. (2009). Evaluation of pairs-trading strategy at the brazilian financial market. *Journal of Derivatives & Hedge Funds*, 15(2):122–136.
- Rad, H., Low, R. K. Y., and Faff, R. (2016). The profitability of pairs trading strategies: distance, cointegration and copula methods. *Quantitative Finance*, 16(10):1541–1558.
- Ramdas, A., Trillos, N. G., and Cuturi, M. (2017). On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47.
- Rapach, D. E., Strauss, J. K., and Zhou, G. (2013). International stock return predictability: What is the role of the united states? *The Journal of Finance*, 68(4):1633–1662.
- Renard, B. and Lang, M. (2007). Use of a gaussian copula for multivariate extreme value analysis: some case studies in hydrology. *Advances in Water Resources*, 30(4):897–912.
- Roll, R. (1984). A simple implicit measure of the effective bid-ask spread in an efficient market. *The Journal of finance*, 39(4):1127–1139.
- Scholz, F. W. and Stephens, M. A. (1987). K-sample anderson–darling tests. *Journal of the American Statistical Association*, 82(399):918–924.
- Sharpe, W. F. (1966). Mutual fund performance. *The Journal of business*, 39(1):119–138.
- Sklar, M. (1959). Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231.
- Sortino, F., van der Meer, R., and Plantinga, A. (1999). The upside potential ratio. *The Journal of Performance Measurement*, 1:10–15.
- Stander, Y., Marais, D., and Botha, I. (2013). Trading strategies with copulas. *Journal of Economic and Financial Sciences*, 6(1):83–107.

- Subrahmanyam, A. (2010). The cross-section of expected stock returns: what have we learnt from the past twenty-five years of research? *European Financial Management*, 16(1):27–42.
- Vayanos, D. and Wang, J. (2009). Liquidity and asset prices: A unified framework. Technical report, National Bureau of Economic Research.
- Vidyamurthy, G. (2004). *Pairs Trading: quantitative methods and analysis*, volume 217. John Wiley & Sons.
- Walker, J. S. (2017). *Fast fourier transforms*. CRC press.
- Xie, W., Liew, R. Q., Wu, Y., and Zou, X. (2016). Pairs trading with copulas. *The Journal of Trading*, 11(3):41–52.
- Xie, W. and Wu, Y. (2013). Copula-based pairs trading strategy. *Asian Finance Association (AsFA)*, 10.