Master's Thesis

A Cluster Factor GARCH Model

Brennan Manning

Student number:	11440457
Date of final version:	July 15, 2021
Master's programme:	Econometrics
Specialisation:	Financial Econometrics
Supervisor:	Ms. H. Li
Second Reader:	Prof. dr. C.G.H. Diks

Faculty of Economics and Business

Amsterdam School of Economics
Faculty of Economics and Business

Abstract

This paper first introduces the key issues in clustering analysis, specifically clustering of time series data, and in multivariate GARCH modelling. Then, a novel multivariate factor GARCH model is introduced, the Cluster Factor GARCH model. This model uses a partitional clustering algorithm to generate representative objects of each cluster which are then used as the factors in the factor GARCH model. Then, the model was applied to applications in risk management and forecasting. The first application was towards Value at Risk forecasting, and a backtest of the Value at Risk found that the Cluster Factor GARCH may potentially be misspecified. The second application was towards forecasting the conditional variance matrix, and the forecasts of the Cluster Factor GARCH model were compared to forecasts from other popular multivariate GARCH models. The Cluster Factor GARCH model was found to perform equivalently to the GOGARCH model and the DCC model.

Statement of Originality

This document is written by Student Brennan Manning who declares to take full responsibility for the contents of this document. I declare that the text and the work presented in this document are original and that no sources other than those mentioned in the text and its references have been used in creating it. UvA Economics and Business is responsible solely for the supervision of completion of the work, not for the contents.

Contents

Chapter 1

Introduction

A major problem in financial analysis and risk management is volatility modelling and volatility forecasting. When the data considered is univariate, this can easily be done with the Autoregressive Conditionally Heteroskedastic (ARCH) model of Engle (1982) and by the Generalized Autoregressive Conditionally Heteroskedastic (GARCH) model of Bollerslev (1986). However, most financial problems are not univariate and can be very high dimensional. To model multivariate volatilities, there have been many generalizations of univariate models to multivariate settings. The most general models are the Vector Error Correction (VEC) model of Bollerslev et al. (1988) and the BEKK model of Engle and Kroner (1995). However, these models are typically too complex to be efficiently estimated in higher dimensional settings. Instead, more attention has been placed on factor models and conditional correlation models. The main focus has been on the Orthogonal model of Alexander (2001) and the Generalized Orthogonal model of van der Weide (2002) for factor models, and the Conditional Conditional Correlation model of Bollerslev (1990) and the Dynamic Conditional Correlation model of Engle (2002) for Conditional Correlation models.

Along with the rise and development of these multivariate GARCH models, machine learning has also increased in usage in financial analysis. Machine learning has mainly been used to predict future values in different markets and to predict defaults in credit systems. Huang et al. (2020) provides an overview of deep learning applications to price predictions and credit default detections. However, unsupervised learning is another paradigm of machine learning which contains the field of cluster analysis, which can be a useful tool in finding groups within data. Finding groups within financial datasets can be a useful tool for portfolio analysis as seen in Tola et al. (2008) and Duarte and De Castro (2020). Cluster analysis has also been useful in factor analysis of financial time series as seen in Begusic and Kostanjcar (2020).

However, while cluster analysis has been used for determining factors, little attention has been focused on modelling volatility with clustering that can directly be used for forecasts. This paper proposes a multivariate factor GARCH model that is based on cluster analysis.

This model is called the Cluster Factor GARCH model and is based on the prototypes which are the representative objects of the clusters. The resulting model is relatively simple as it only relies on univariate GARCH estimation and maximum likelihood estimation of loading parameters. However, for particular clustering algorithms, mainly fuzzy clustering algorithms, the estimation of the factor loadings matrix can be omitted to improve computational efficiency.

An important part of financial time series analysis is the ability of a model to accurately forecast a model. To test the performance and accurate specification of the Cluster Factor GARCH model, the model is used to generate a Value at Risk series which is then backtested to test the models specification and forecasting ability. The Value at Risk backtest is the Generalized Method of Moments test of Candelon et al. (2011) which implements a duration based test that has favorable power properties over similar duration based tests and Likelihood Ratio based tests. Then, the model's forecasting ability is compared to other multivariate GARCH models. West (2006) provides an overview of testing forecast performance based on a pairwise comparison of predictive ability. However, this can become complicated in multivariate settings and when a large number of models is considered. Pairwise comparisons also require a benchmark model to be chosen that can complicate inference. Instead, the focus in this case will be based on the Model Confidence Set of Hansen et al. (2011b) that allows for testing which models out of some large set of models are able to produce the most accurate forecasts. This framework is easily extended to handle multivariate GARCH models. The Cluster Factor GARCH model is thus compared to the GOGARCH Model, the DCC model, and BEKK type models for forecasts of horizons of 1, 5, and 20 days based on the Model Confidence Set approach.

The structure of the rest of the paper is as follows. Chapter 2 provides an overview of related works on clustering in finance and multivariate GARCH models. Chapter 3 provides a brief overview of relevant clustering theory and algorithms. Chapter 4 provides an overview of univariate GARCH models and multivariate GARCH models including the theory on estimation and forecasting. Chapter 5 introduces the Cluster Factor GARCH model and its estimation and forecasting procedures. Chapter 6 presents an empirical application of the Cluster Factor GARCH and reports the results of the Value at Risk backtest and the Model Confidence Set test. Chapter 7 concludes.

Chapter 2

Literature Review

2.1 Clustering in Finance

There have been many recent papers using clustering algorithms in order to find groups and potentially find representative objects. Begusic and Kostanjcar (2020) presents a factor model for asset returns that contains pervasive factors and factors that are cluster specific. The method of finding the pervasive factors is by principal components, while finding the cluster specific components depends on a spectral clustering method based on the Laplacian matrix of a graph whose connections are based on correlations. The authors conclude that their method is able to outperform a standard principal component factor model in explaining out-of-sample variance and allows for better forecasting results than if you were to rely only on the principal components results. Verma et al. (2019) also constructs a factor model based on clustering in order to try to explain the volatility clustering in stock returns. The clustering in this case is done by a hierarchical method based on the correlations between stocks. The authors use a proxy for volatility being the logarithm of the returns and build their volatility model based on a factor model including a market model. The factors are built as a weighted average of each member in a cluster.

In addition to building factor models from clusters, clustering can also aid in constructing portfolios. Tola et al. (2008) uses clustering to find a simplified correlation matrix from the large dimension correlation matrix which the authors then use to find the optimal Markowitz portfolio. Duarte and De Castro (2020) uses partitional clustering methods to initially find clusters in asset returns based on correlations. With the clusters found, they propose a method for asset allocation, first by allocating funds first to each cluster and then to allocating those funds to the assets within each cluster. The authors found that on a return basis their method was able to outperform a Markowitz based portfolio and other indices.

2.2 Multivariate GARCH Models

For a survey of most available multivariate GARCH models, Bauwens et al. (2006) and Silvennoinen and Teräsvirta (2009) both offer an introduction to the theory behind most multivariate GARCH models and their properties. Both papers explain the most popular multivariate GARCH models and popular extensions of those models. Silvennoinen and Teräsvirta (2009) also offers an application of such models to an empirical example and also provides information on semi-parametric estimation of multivariate GARCH models. Laurent et al. (2012) does an extensive test of 125 multivariate GARCH models for forecasting performance by the Model Confidence Set of Hansen et al. (2011b) and the Superior Predictive Ability test of Hansen (2005). The authors use various methods to quantize the performance, but they ultimately find that the Orthogonal GARCH model and the DCC type models are more likely to outperfrom other types of models such as BEKK or VEC models. They also found that introducing leverage terms in modelling the returns of the univariate processes increases the forecasting performance of the DCC and OGARCH models.

Another novel multivariate factor GARCH model was that of Santos and Moura (2014). The authors introduce the Dynamic Factor GARCH (DFGARCH) model which allows for time varying factor loadings under the restriction that factors must be observed. The DFGARCH model allows for the factor covariance matrix to be estimated by a DCC model and not restricted to be diagonal. Then, a state space model is used to estimate the factor loadings. The authors showed that when considering a portfolio optimization problem that their model was able to outperform other currently available models.

Chapter 3

Cluster Analysis

The goal of clustering is to group together observations in a dataset such that each observation is placed with the observations that are most similar to it. In particular, the groups, known formally as clusters, are made so that each observation will be most similar to the observations in the same cluster and least similar to all observations in the other clusters. This technique can be used to identify a structure in the data that can then be utilized in other tasks. Formally, given a dataset $\mathcal{D} = \{x_1, \ldots, x_n\}$, the goal of clustering will be to partition $\mathcal{D}$ into a collection of disjoint subsets $\{C_1, \ldots, C_k\}$ so that $\mathcal{D} = \bigcup_{i=1}^{k} C_i$ and $C_i \bigcap C_j = \emptyset$ for $i \neq j$.

In order to begin clustering, the notion of similarity or dissimilarity needs to be defined. Kaufman and Rousseeuw (2009) defines a dissimilarity measure $d(x_i, x_j)$ between observations x_i and x_j to be a non-negative number which is close to 0 if x_i and x_j are in some sense near to each other and will be larger if x_i and x_j are very different from each other. A similarity measure is defined similarly however, similar observations will have a higher measure than dissimilar observations. Then, the only remaining choice is for the function d to measure the dissimilarity between observations. An important assumption to be made for the function d is that it should be symmetric, so that $d(x_i, x_j) = d(x_j, x_i)$. A typical choice for this would be any distance function such as the Euclidean distance or the Manhattan distance, but general distance functions also work. A common similarity measure is the correlation coefficient. To use the a similarity measure in place of a dissimilarity measure requires the measure to be transformed. The two most common transformations are

$$d(x_i, x_j) = 1 - |\rho_{i,j}| \quad \text{or} \quad d(i, j) = \frac{1}{2}(1 - \rho_{i,j})$$

where $\rho_{i,j}$ is the correlation coefficient between x_i and x_j. The choice of which transformation to use may depend on the use case as it may depend on whether negative correlation should be considered similar or dissimilar.

After the choice of dissimilarity has been made, the next choice in cluster analysis is the type of clustering algorithm to be used. Saxena et al. (2017) provides a review of the most common clustering algorithms and classifies them into two main classes: hierarchical and partitional.

Hierarchical methods attempt to form clusters by iteratively dividing the observations or iteratively combining observations. Hierarchical methods however, are not the focus of this paper and most focus is on their counterpart, partitional methods. Partitional methods instead assign each observation to one of k clusters based on some criterion function. The most prominent of these methods is the k-means algorithm. The main idea behind the k-means algorithm, is that there are k clusters and each cluster has a centroid which in this case is the mean of all observations in that cluster. Saxena et al. (2017) provides the criterion function

$$\min J = \sum_{j=1}^{k} \sum_{i=1}^{n} d(x_i^{(j)}, c_j)$$

where $x_i^{(j)}$ is x_i if the observation belongs to the j-th cluster and c_j is the centroid of the j-th cluster. The algorithm for k-means is essentially composed of two steps. The first step is to assign each observation to the closest centroid and the second step is to recalculate the centroid to be the average of all observations in that cluster. These two steps are then repeated until the centroids converge and the clusters remain the same. However, a problem in the algorithm is that the resulting clustering will depend on the initial centroids chosen. A typical way to counteract this is to randomly initialize the centroids and run the algorithm multiple times with different initializations.

Another clustering problem is k-medoids which is similar to k-means. However, instead of constructing centroids from each cluster, k observations need to be found that are representative of each of the k clusters which are then called the medoids. After finding the k representative observations, each observation is then assigned to the cluster with the nearest medoid. The algorithm to solve this problem is the Partition Around Medoids(PAM) algorithm which was developed by Kaufman and Rousseeuw (2009).

The above methods discussed have all been traditional, or crispy, clustering algorithms. However, there is another type of clustering, namely fuzzy clustering. Fuzzy clustering allows for observations to belong to multiple clusters with a degree of membership, while crispy clustering forces each observation into a single cluster. The most well known fuzzy clustering algorithm is the fuzzy counterpart to k-means, fuzzy c-means. Saxena et al. (2017) provides the following minimization problem for fuzzy c-means

$$\min J = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^{m} d(x_i, v_j)$$

$$\text{s.t.} \sum_{j=1}^{c} u_{ij} = 1$$

where u_{ij} is the degree of membership of observation x_i to cluster j, m is the fuzzifier exponent where $1 < m < \infty$, and v_j is the centroid of cluster j. m is a constant that must be chosen by the researcher and will impact the degree of "fuzziness" where low values more closely resemble the results of a crispy algorithm and high values are more "fuzzy".

3.1 Time Series Clustering

While the above section lays out some of the general theory about clustering, the following section will lay out some of the differences one has to deal with when clustering time series data. Aghabozorgi et al. (2015) discusses some of the differences between general clustering and time series clustering and also provides an introduction to time series clustering. One main difficulty in time series clustering is that a dataset can have observations that will have thousands of datapoints and can be expensive to do intensive computations on the dataset. However, a way to overcome this is by transforming the time series to a different equivalent representation. Aghabozorgi et al. (2015) defines for a time series $X_i = (x_1, \ldots, x_T)$, a representation of this time series X_i is the transformation to a dimension reduced vector $X_i' = (x_{i1}', \ldots, x_{iU}')$ where $U < T$. A representation of time series should not affect the similarity between any two time series, so that if two time series are similar in the original space, they should remain similar in the transformed space.

There are numerous ways of transforming a time series with some of the most common being the Wavelet Transformation and the Fourier Transformation. However, a relatively simple transformation, the Pieceweise Aggregate Approximation (PAA), was introduced by Keogh and Pazzani (2000b). The PAA representation takes a time series X_i of length T and transforms it into a time series $\bar{X}_i = (\bar{x}_{i1}, \ldots, \bar{x}_{iU})$ of length U where each element of $\bar{X}_i$ is calculated as

$$\bar{x}_{i,t} = \frac{U}{T} \sum_{j=\frac{U}{T}(t-1)+1}^{\frac{U}{T}t} x_{i,j}.$$

The PAA representation of a time series can then be used in the clustering algorithms instead of the raw time series which can ease the memory cost of the data and can allow for quicker computations of the dissimilarity between time series.

After determining the representation for the time series, a necessary component is then the dissimilarity measure between each time series. As in traditional time series, correlation and distance functions can be used as a dissmilarity measure. However, due to the temporal ordering of the data, there can be more information extracted from the indexing of the data and more interesting distance measures can be used. One of the most common distance measures that can be used is the Dynamic Time Warping (DTW) distance which was originally developed for speech recognition by Sakoe (1971) and was introduced to time series problems by Berndt and Clifford (1994). The DTW algorithm is a dynamic programming algorithm which tries to find a warping path between two time series such that the distance followed by the warping path is minimized. However, since the problem is solved by dynamic programming, it can be quite expensive to calculate between two pairs of long time series and might become infeasible to calculate when considering a large number of time series. Keogh and Pazzani (2000a) finds that using the PAA transformation in conjunction with DTW in a clustering algorithm can provide

faster computations while also retaining some of the benefits of DTW when compared to using Euclidean distance.

With the representation and dissimilarity measure decided, the choice of clustering algorithm is the next choice. Compared to traditional clustering, k-means may no longer be the most convenient choice for clustering, as the definition of the average time series in a cluster may become difficult in computing especially when considering different representations and non-Euclidean distances. Instead, the k-medoid approach may be a more favorable technique for this problem as the prototypes no longer need to be calculated and can be found in the data. Fortunately, the PAM algorithm can also be used in time series clustering to solve the k-medoid problem. The original algorithm by Kaufman and Rousseeuw (2009) only requires a dissimilarity matrix to find the k central medoids. A fuzzy extension of k-medoids that can be easily used in time series is the Fuzzy c-Medoids (FCMdd) algorithm that was initially developed by Krishnapuram et al. (2001). The algorithm for time series was then demonstrated by Izakian et al. (2015) and is shown to minimize the following criterion function

$$J = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^{m} d(x_i, v_j)$$

where again u_{ij} is the degree of membership of observation x_i to cluster j, m is the fuzzifier exponent, and v_j is the medoid of cluster j. The algorithm to solve this minimization problem is then to initially select c time series randomly to be the first c medoids. Then, calculate the membership of each of the time series to each of the c clusters as follows

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d(x_i, v_j)}{d(x_i, v_k} \right)^{2/(m-1)}}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, c$. Then, the next step is to calculate the c most central medoids in each of the clusters which is done as follows

$$v_j = X_q$$
$$q = \arg \min_{1 \leq i \leq n} \sum_{j=1}^{n} d(x_i, x_j).$$

This process of calculating the membership and finding the central medoids is repeated until either a number of maximal iterations is met or until the chosen medoids do not change.

Chapter 4

Multivariate GARCH Models

A key issue in time series analysis, especially for financial time series, is the modelling of the volatility of the series. A common way to do this is with a GARCH model for either univariate or multivariate series. The notation in the following sections is then as follows. $\{r_t\}_{t=1}^T$ will be a series of returns for either a single asset or for multiple assets. μ_t will denote the conditional mean of r_t with respect to the σ-field $\mathcal{F}_{t-1} = \sigma(r_1, \ldots, r_{t-1})$. $\{a_t\}_{t=1}^T$ will represent the series of innovations for the return series. h_t will represent the conditional variance of a_t with respect to the σ-field $\mathcal{F}_{t-1}$ when considering only a single asset and H_t will represent the conditional variance matrix for a_t with respect to the σ-field $\mathcal{F}_t$ when considering multiple assets.

4.1 Review of Univariate GARCH Models

GARCH models are the generalization of the earlier ARCH model. The framework for a GARCH(p,q) model is the following

$$\begin{aligned} r_t &= \mu_t + a_t \\ a_t &= \sqrt{h_t} z_t \\ h_t &= \alpha_0 + \sum_{i=1}^{p} \alpha_i a_{t-i}^2 + \sum_{j=1}^{q} \beta_j h_{t-j} \\ \alpha_0 &> 0, \alpha_i \geq 0, \beta_j \geq 0 \end{aligned}$$

where z_t is a series of i.i.d. standard random variable, typically standard normal or the standardized Student's t distribution, and the constraints are to ensure that the resulting values of h_t are positive. Typically, however, a GARCH(1,1) model can adequately model the observed innovation series without the need for higher terms. The GARCH(1,1) model models the conditional variance as follows

$$\begin{aligned} h_t &= \alpha_0 + \alpha_1 a_{t-1}^2 + \beta_1 h_{t-1} \\ \alpha_0 &> 0, \alpha_1 \geq 0, \beta_1 \geq 0. \end{aligned}$$

A special case of the GARCH(1,1) model is the integrated GARCH model, IGARCH(1,1), where $\alpha_1+\beta_1 = 1$. Morgan (1996) further simplifies the IGARCH(1,1) model by assuming that $\alpha_0 = 0$ and that $(1-\alpha_1) = \beta_1 = 0.94$ so that the RiskMetrics model of the conditional variance is then

$$h_t = 0.06r_{t-1}^2 + 0.94h_{t-1}$$

There are also numerous extensions to the standard GARCH model such as the exponential GARCH (EGARCH) model of Nelson (1991) and the GJR-GARCH model of Glosten et al. (1993). These models aim to allow for the leverage effect to be modelled in the series where the leverage effect implies that a large negative shock is expected to have a larger impact on volatility when compared to a large positive shock. The EGARCH(1,1) model models this by modelling the logarithm of the conditional volatility by

$$\log h_t = (1-\alpha_1)\alpha_0 + \theta z_{t-1} + \gamma(|z_{t-1}| - \mathbb{E}[|z_{t-1}|]) + \alpha_1 \log h_{t-1}.$$

All of the above GARCH models above are estimated by maximum likelihood. The distribution of the model depends on the choice of distribution for the standard i.i.d. random variables $\{z_t\}_{t=1}^T$. All of the estimation results for this section will be with the assumption that each z_t follows the standard normal distribution. The likelihood and log-likelihood function for the model are as follows

$$\begin{aligned} L(\phi, \psi; r_1, \ldots, r_T) &= \prod_{t=2}^{T} p(r_t; \phi, \psi | \mathcal{F}_{t-1}) \\ \log L(\phi, \psi | r_1, \ldots, r_T) &= \sum_{t=2}^{T} \log p(r_t; \phi, \psi | \mathcal{F}_{t-1}) \\ &= \sum_{t=2}^{T} \left[-\frac{1}{2}\log 2\pi - \frac{1}{2}\log h_t(\psi) - \frac{1}{2}\left(\frac{(r_t - \mu_t(\phi))^2}{h_t(\psi)} \right) \right] \end{aligned}$$

where ϕ is the parameter vector for the estimation for the conditional mean and ψ is the parameter vector for the parameters in the GARCH type model Additionally, in the above equation it was assumed that both the conditional mean and conditional variance were assumed to only depend on at most one previous observation.

4.2 Multivariate GARCH Models

Univariate GARCH models are typically only enough to model the volatility of a single asset. However, in many applications, such as portfolio construction and risk management, one would want the volatility of multiple assets and the covariance dynamics between each asset of interest. To do this, the multivariate generalization of the univariate GARCH models can be used. A

more detailed explanation of all these types of models is given by Bauwens et al. (2006) and Silvennoinen and Teräsvirta (2009).

Now, the focus of $\{r_t\}_{t=1}^T$ is when $r_t = (r_{1,t}, \ldots, r_{N,t})'$ where N is the number of series considered. The goal of a multivariate GARCH model is to estimate the following conditional variance matrix

$$H_t = \text{Var}[a_t|\mathcal{F}_{t-1}] = \mathbb{E}[a_t a_t'|\mathcal{F}_{t-1}].$$

A problem that most multivariate GARCH models face is that they must be flexible enough to accurately model the volatilities and covariances, while still remaining feasible to estimate in high dimensional applications.

4.2.1 VEC and BEKK Models

The most general multivariate GARCH model is the Vector Error Correction Model (VEC) model. The VEC(1,1) model is defined as

$$\text{vech } H_t = c + A\text{vech}(a_{t-1}a_{t-1}') + B\text{vech } H_{t-1}$$

where vech$A$ is the half-vectorization of the matrix A, c is a $N^* \times 1$ vector, A and B are both $N^* \times N^*$ matrices, and $N^* = 1/2N(N+1)$. The main drawback of this model is that the number of parameters grows as $\mathcal{O}(N^4)$ and will be infeasible to estimate for large values of N. Instead, a way to simplify this is to assume that A and B are both diagonal matrices which leads to the diagonal VEC, DVEC, model which for an order (1,1) model is defined by

$$H_t = C + A \odot (a_{t-1}a_{t-1}') + B \odot H_{t-1}$$

where $\odot$ is the Hadamard product, and C, A, and B are all $N \times N$ matrices. A special case of this is the scalar VEC model, and in particular the multivariate RiskMetrics model, which is then given by

$$H_t = (1-\lambda)a_{t-1}a_{t-1}' + \lambda H_{t-1}$$

where for the RiskMetrics model $\lambda = 0.94$. Another drawback to VEC type models is that there are numerous conditions on the parameter matrices to ensure that H_t will be semi-positive definite which can make estimation onerous. Instead, Engle and Kroner (1995) proposes the BEKK model which is a different parametrization of H_t that ensures positive definite at the cost of flexibility. The BEKK(1,1,K) model is defined by

$$H_t = C'C + \sum_{k=1}^{K} A_k' a_{t-1}a_{t-1}' A_k + \sum_{k=1}^{K} B_k' H_{t-1} B_k$$

where C is a lower triangular $N \times N$ matrix, and each A_k and B_k are non-singular $N \times N$ matrices. The most common BEKK model is the BEKK(1,1,1) model which is given by

$$H_t = C'C + A'a_{t-1}a_{t-1}'A + B'H_{t-1}B.$$

This model is more easily estimated compared to a VEC model as it will only have $\mathcal{O}(N^2)$ parameters that need to be estimated. However, it can still be infeasible to estimate for a large number of series. One can instead estimate scalar BEKK and diagonal BEKK models which have a similar form to scalar VEC and diagonal VEC models, however the BEKK models will be guranteed to be positive definite. The scalar BEKK model will have C defined the same and A and B will be diagonal matrices such that $A = aI_N$ and $B = bI_N$ with I_N being the $N \times N$ identity matrix and the diagonal BEKK model will have A and B being diagonal matrices where each diagonal element needs to be estimated.

4.2.2 Factor Models

While VEC and BEKK models try to directly model the GARCH structure of the series, factor models instead try to model the volatility based off of other series, known as factors. This sort of analysis is known as factor analysis. Factor models are common in financial models as most financial data is typically high dimensional and factor analysis can substantially reduce the dimensionality of the data and the number of parameters in the model. The basic form for a factor GARCH model is

$$a_t = Gf_t + \varepsilon_t$$

where f_t are the factors, G are the factor loadings and ε_t is an error term. The factors $f_t = (f_{1,t}, \ldots, f_{k,t})'$ are assumed to be independent GARCH processes so that $\text{Cov}[f_{i,t}, f_{j,t}] = 0 \quad \forall i \neq j$, and they are assumed to be independent of the error term ε_t so that $\mathbb{E}[f_{i,t}\varepsilon_{j,t}] = 0$ for $i = 1, \ldots, k$ $j = 1, \ldots, N$. The error terms ε_t are assumed to be i.i.d. with mean 0 and variance Ω. The conditional variance implied by this model is then

$$\begin{aligned} H_t &= G\Sigma_t G' + \Omega \\ \Sigma_t &= \text{Var}[f_t|\mathcal{F}_{t-1}] = \text{diag}(\sigma^2_{1,t}, \ldots, \sigma^2_{k,t}) \end{aligned}$$

where the variance matrix of the factors is diagonal since the factors are assumed to be independent and $\sigma_{i,t}$ is the conditional volatility of the i-th factor.

A specific factor GARCH is the Orthogonal GARCH (OGARCH) model developed by Alexander (2001). The basis behind the OGARCH model is that the factors can be generated by principal component analysis of the unconditional correlations of the innovations a_t. The first m principal components are then chosen to be the factors in the model. The OGARCH model without an error term can then be formulated as

$$\begin{aligned} a_t &= Gf_t \\ H_t &= G\Sigma_t G' \end{aligned}$$

where

$$
\begin{aligned}
G &= D\begin{pmatrix} e_1 & \dots & e_m \end{pmatrix}\Lambda^{1/2} \\
f_t &= \Lambda^{-1/2}\begin{pmatrix} e_1 & \dots & e_m \end{pmatrix}D^{-1}a_t \\
D &= \operatorname{diag}(\sqrt{h_{1,1}}, \dots, \sqrt{h_{N,N}}) \\
\Lambda &= \operatorname{diag}(\lambda_1, \dots, \lambda_m)
\end{aligned}
$$

where $\lambda_1 \geq \cdots \geq \lambda_N$ are the eigenvalues of $\operatorname{Corr}[a_t] = D^{-1}HD^{-1}$ with $H = \operatorname{Var}[a_t]$, $\{e_i\}_{i=1}^N$ are the corresponding eigenvectors, and $\{h_{i,i}\}_{i=1}^N$ are the diagonal elements of H. The model is simple to estimate as it only relies on principal component analysis and univariate GARCH estimation. A downfall of this model is that the resulting conditional variance matrix will be singular if $m < N$. Another shortcoming of the OGARCH model, is that if the factors have similar unconditional variances, then the model will have indentification problems. van der Weide (2002) proposes a way to fix this by introducing conditional information into the estimation of the factor loadings. This model is then known as the generalized orthogonal GARCH model (GOGARCH). The GOGARCH model introduces an orthogonal matrix into the structure of the factor loadings matrix. The GOGARCH model can then be formulated as

$$
\begin{aligned}
a_t &= Gf_t \\
G &= DE\Lambda^{1/2}U \\
E &= \begin{pmatrix} e_1 & \dots & e_N \end{pmatrix} \\
\Lambda &= \operatorname{diag}(\lambda_1, \dots, \lambda_N)
\end{aligned}
$$

where D, f_t, and the eigenvalue-eigenvector pairs are defined in the same way as in the OGARCH model, and U is some orthogonal matrix. The estimation of the GOGARCH model can be similar to that of the OGARCH model.

4.2.3 Correlation Models

An alternative approach to modelling the conditional variance matrix is to instead model the correlations of the series over time instead of linear combinations of the variances and series over time. A model of the correlations is then a non-linear combination of the individual variances and series of the multivariate series considered.

The first model is the constant conditional correlation (CCC) model of Bollerslev (1990). The CCC model assumes that the correlations between individual series are constant over time.

The CCC model is then given by

$$H_t = D_t R D_t$$
$$D_t = \text{diag}(\sqrt{h_{1,t}}, \ldots, \sqrt{h_{N,t}})$$

where $R = \text{Corr}[a_t]$ and $\{h_{i,t}\}_{i=1}^N$ are the conditional volatilities of the individual series which are typically modelled by some univariate GARCH model. This model is simple to implement with only the correlation matrix and the univariate GARCH models needing to be estimated, however it can be very restrictive to assume that the correlations between series are constant over time.

The Dyanmic Conditional Correlation (DCC) of Engle (2002) model attempts to remove the constant correlation restriction of the CCC model by allowing the correlations to vary over time. The DCC model is then given by

$$H_t = D_t R_t D_t$$
$$R_t = (Q_t^*)^{-1} Q_t (Q_t^*)^{-1}$$
$$Q_t = (1 - a - b)\bar{Q} + a z_{t-1} z_{t-1}' + b Q_{t-1}$$

where $Q_t^* = \text{diag}(\sqrt{q_{1,1,t}}, \ldots, \sqrt{q_{N,N,t}})$, $q_{i,i,t}$ are the diagonal elements of the matrix Q_t, $z_t = D_t^{-1} a_t$, and $\bar{Q} = \text{Var}[z_t]$.

4.3 Estimation

Estimation of multivariate GARCH models is typically done with maximum likelihood. The most common distributions used for this are the multivariate normal, multivariate Student's t, and the generalized error distribution. As in the univariate case, however, the focus in this paper will be solely on the multivariate normal case. The general form for maximum likelihood estimation of the different types of models will be

$$\log L(\theta; r_t) = \sum_{t=2}^{T} \ell_t(\theta)$$
$$\ell_t = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|H_t(\theta)| - \frac{1}{2}(r_t - \mu_t(\theta))' H_t(\theta)^{-1}(r_t - \mu_t(\theta)$$

where θ is the parameter vector for both the mean process and the variance process. By Quasi-maximum likelihood properties, the estimated parameters will be consistent. Maximum likelihood estimation is used to estimate the parameters for the BEKK and factor models. However, this type of estimation can be burdensome especially for higher dimension series. Instead, in the GOGARCH and DCC models, different estimation methods are possible.

4.3.1 Factor Models

The estimation of the OGARCH model of Alexander (2001) can be estimated by using the sample estimate of the unconditional variance of the innovations a_t, $\hat{\text{Cov}}[a_t] = 1/T \sum_{i=1}^{T} a_i a_i'$, to estimate the unconditional correlation denoted as $\hat{R}$ and then perform principal components analysis on $\hat{R}$. The principal component analysis then gives $\hat{R} = E\Lambda E'$ where E is the matrix of eigenvectors and Λ is a diagonal matrix of eigenvalues. With this the factors can be constructed with

$$f_t = \Lambda^{-1/2} E' a_t.$$

Then, for the first m factors chosen, $f_{i,t}$, a univariate GARCH model can be estimated to estimate the conditional variance matrix using the general form for a factor model as given in section 4.2.2.

The estimation procedure for the GOGARCH model can be similar to the procedure used to estimate the OGARCH model. The main addition is that the orthogonal matrix U needs to estimated. This can be done by performing the principal components as above and then optimizing the log-likelihood function of this model over orthogonal matrices U. The likelihood function as given by van der Weide (2002) is as follows

$$\log L(\theta; a_t) = -\frac{1}{2} \sum_{t=1}^{T} N \log 2\pi + \log |E\Lambda E'| + \log |\Sigma_t| + f_t' \Sigma_t^{-1} f_t$$

where $\Sigma_t = \mathbb{E}[f_t f_t' | \mathcal{F}_{t-1}]$ and this can be optimized for the parameters in the orthogonal matrix U, which arises from f_t, and the univariate GARCH parameters used to model the factors' variance. However, there are alternative estimation procedures to estimate the GOGARCH model. Boswijk and van der Weide (2011) proposes a method of moments estimation procedure to estimate the orthogonal matrix U and uses a polar decomposition instead of principal component analysis to define the factors. Another estimation method for the GOGARCH model is by Zhang and Chan (2009), which they denote as the independent factors GARCH model, where the authors replace principal components analysis by Independent Components Analysis.

4.3.2 Correlation Models

The likelihood of the DCC model can broken into two parts which Engle (2002) refers to as the volatility component and the correlation component. The full log-likelihood function for the DCC model is given by

$$\begin{aligned} \log L(\theta; r_t) &= \log L_v(\theta_1; r_t) + \log L_c(\theta_2; r_t) \\ &= -\frac{1}{2} \sum_{t=1}^{T} N \log 2\pi + 2 \log |D_t| + a_t' D_t^{-1} D_t^{-1} a_t - z_t' z_t + \log |R_t| + z_t' R_t^{-1} z_t \end{aligned}$$

where L_v is the likelihood of the volatility component, L_c is the likelihood of the correlation component, and $z_t = D_t^{-1} a_t$. The log-likelihood functions of the volatility and correlation

components are as follows

$$\begin{aligned}\log L_v(\theta_1; r_t) &= -\frac{1}{2}\sum_{t=1}^{T} N\log 2\pi + \log|D_t|^2 + a_t' D_t^{-2} a_t \\ &= -\frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{N} \log 2\pi + \log|h_{i,t}| + \frac{a_{i,t}^2}{h_{i,t}} \\ \log L_c(\theta_2; r_t) &= -\frac{1}{2}\sum_{t=1}^{T} \log|R_t| + z_t' R_t^{-1} z_t - z_t' z_t\end{aligned}$$

where $h_{i,t}$ is the conditional variance of the i-th series. The second equation for the log-likelihood of the volatility component can be seen as the sum of likelihood functions for each of the individual series and thus can be optimized by optimizing each individual likelihood. The procedure for optimization is then to estimate a univariate GARCH model for each individual time series in r_t, and then optimize the correlation component using the estimated standard deviation matrix $\hat{D}_t$ to calculate z_t. Engle (2002) shows that this method will be consistent if the first stage is consistent, however they also show that this method will be inefficient compared to optimizing the full log-likelihood function.

4.4 Forecasting

A key part of volatility analysis is the ability to generate accurate forecasts of the volatility at certain points in the future. This can play a crucial role in portfolio optimization and risk management specifically analyzing the Value at Risk and the Expected Shortfall. The main idea behind forecasting is to estimate what is expected to occur k days ahead from the present. To do this, the expectation of the model is considered given the information set up to the present. The k day ahead forecast of the variance matrix, $\hat{H}_{t+k}$, is then

$$\hat{H}_{t+k} = \mathbb{E}[a_{t+k} a_{t+k}' | \mathcal{F}_t].$$

4.4.1 Univariate GARCH

Most of the multivariate GARCH models forecasts' depend on the forecasts of their univariate GARCH components. So, in order to show the forecasts of the multivariate GARCH models, the univariate GARCH forecasts must first be developed. The starting point is on the GARCH(1,1) model. If a_t follows a univariate GARCH(1,1) model with conditional variance h_t, then h_{t+1} will be defined as follows

$$h_{t+1} = \alpha_0 + \alpha_1 a_t^2 + \beta_1 h_t.$$

The one day ahead forecast for a GARCH(1,1) model will then be

$$\begin{aligned}\hat{h}_{t+1} &= \mathbb{E}[h_{t+1}|\mathcal{F}_t] = \mathbb{E}[\alpha_0 + \alpha_1 a_t^2 + \beta_1 h_t|\mathcal{F}_t] \\ &= \alpha_0 + \alpha_1 a_t^2 + \beta_1 h_t.\end{aligned}$$

Then, for larger horizon forecasts when $\ell > 1$, the forecasts have a recursive nature which is shown as follows

$$\begin{aligned}\hat{h}_{t+\ell} &= \alpha_0 + \mathbb{E}[\alpha_1 a_{t+\ell-1}^2 + \beta_1 h_{t+\ell-1}|\mathcal{F}_t] \\ &= \alpha_0 + \alpha_1 \mathbb{E}[a_{t+\ell-1}^2|\mathcal{F}_t] + \beta_1 \mathbb{E}[h_{t+\ell-1}|\mathcal{F}_t] \\ &= \alpha_0 + (\alpha_1 + \beta_1)\mathbb{E}[h_{t+\ell-1}|\mathcal{F}_t] = \alpha_0 + (\alpha_1 + \beta_1)\hat{h}_{t+\ell-1}.\end{aligned}$$

Similarly, for the EGARCH(1,1) model a recursive forecast structure can be given as follows

$$\hat{h}_{t+\ell} = \begin{cases} h_t^{\beta_1} \exp\{\omega + \theta z_t + \gamma(|z_t| - \mathbb{E}[|z_t|])\} & \ell = 1 \\ \hat{h}_{t+\ell-1}^{\beta_1} \exp\{\omega + \mathbb{E}[\theta z + \gamma(|z| - \mathbb{E}[z])]\} & \ell > 1 \end{cases}$$

where for the case when $\ell > 1$ the fact that z_t is an i.i.d. series of standard random variables is used and the evaluation of the expectation in this case will depend on the chosen distribution.

4.4.2 BEKK

The forecasts of the BEKK models, however do not rely on univariate GARCH models and will instead rely on a recursive formulation for the ℓ day ahead forecasts. The one day ahead forecast of a BEKK(1,1,1) model is then

$$\begin{aligned}\hat{H}_{t+1} &= \mathbb{E}[a_{a+1} a_{t+1}'|\mathcal{F}_t] \\ &= C'C + A' a_t a_t' A + B' H_t B\end{aligned}$$

which can then be used in the following recursive formulation for the general ℓ day ahead forecast of the BEKK(1,1,1) model

$$\hat{H}_{t+k} = C'C + (A+B)' \hat{H}_{t+\ell-1}(A+B) \quad k > 1.$$

4.4.3 Factor Models

The forecasts of factor models, including the OGARCH model and GOGARCH model, is composed of the univariate GARCH forecasts and the factor loadings matrix which is constant over time. The ℓ day forecast for a general factor GARCH model is then

$$\begin{aligned}\hat{H}_{t+\ell} &= G\hat{\Sigma}_{t+\ell} G' + \Omega \\ \hat{\Sigma}_{t+\ell} &= \operatorname{diag}(\hat{\sigma}_{1,t+\ell}^2, \ldots, \hat{\sigma}_{k,t+\ell}^2)\end{aligned}$$

where $\hat{\sigma}_{i,t+\ell}$ is the ℓ day ahead forecast of the volatility of the i-th factor based on the univariate GARCH specification of that factor.

4.4.4 Correlation Models

The first stage in the forecast procedure for the DCC model is similar to the procedure for factor models. The first stage is to forecast the diagonal matrices of conditional variances D_t which can be accomplished by

$$\hat{D}_{t+\ell} = \text{diag}(\sqrt{\hat{h}_{1,t+\ell}}, \dots \sqrt{\hat{h}_{N,t+\ell}})$$

where $\hat{h}_{i,t+\ell}$ is the ℓ day ahead forecast of the volatility of the i-th series. In contrast, forecasting the correlation matrix R_t is more difficult which is based on the forecast of the matrix Q_t. The ℓ day ahead forecast of Q_t will be

$$\mathbb{E}[Q_{t+\ell}|\mathcal{F}_t] = \begin{cases} (1-a-b)\bar{Q} + az_t z_t' + bQ_t & \ell = 1 \\ (1-a-b)\bar{Q} + a\mathbb{E}[z_{t+\ell-1}z_{t+\ell-1}|\mathcal{F}_t] + b\mathbb{E}[Q_{t+\ell-1}|\mathcal{F}_t] & \ell > 1 \end{cases}$$

where $\mathbb{E}[z_{t+\ell-1}z_{t+\ell-1}'|\mathcal{F}_t] = \mathbb{E}[R_{t+k-1}|\mathcal{F}_t] = \mathbb{E}[(Q^*_{t+\ell-1})^{-1}Q_{t+\ell-1}(Q^*_{t+\ell-1})^{-1}|\mathcal{F}_t]$. This quantity is unknown and needs to be approximated. Sheppard and Engle (2001) proposes two methods to estimate this quantity. The first method they proposed was to assume that $\mathbb{E}[z_{t+\ell}z_{t+\ell}|\mathcal{F}_t] \approx \mathbb{E}[Q_{t+\ell}|\mathcal{F}_t]$ which leads to the following forecast for $Q_{t+\ell}$

$$\hat{Q}_{t+\ell} = (1-(a+b)^{\ell-1})\bar{Q} + (a+b)^{\ell-1}\hat{Q}_{t+1}$$

which can be used to get the forecast of the correlation matrix with $\hat{R}_{t+\ell} = (\hat{Q}^*_{t+\ell})^{-1}\hat{Q}_{t+\ell}(\hat{Q}^*_{t+\ell})^{-1}$ with $\hat{Q}^*_{t+\ell}$ being a diagonal matrix consisting of the square roots of the diagonal elements of $\hat{Q}_{t+k}$. The second method they propose was to assume that $\bar{Q} \approx \bar{R}$ and that $\mathbb{E}[R_{t+\ell}|\mathcal{F}_t] \approx \mathbb{E}[Q_{t+\ell}]$. This leads to the following forecast for $R_{t+\ell}$

$$\hat{R}_{t+\ell} = (1-(a+b)^{\ell-1})\bar{R} + (a+b)^{\ell-1}\hat{R}_{t+1}$$

with $\hat{R}_{t+1} = (\hat{Q}^*_{t+1})^{-1}\hat{Q}_{t+1}(\hat{Q}^*_{t+1})^{-1}$. Sheppard and Engle (2001) found that the second method was less biased and lead to more accurate forecasts. The final forecast for the DCC model, regardless of the method chosen to approximate $R_{t+\ell}$, will then be

$$\hat{H}_{t+\ell} = \hat{D}_{t+\ell}\hat{R}_{t+\ell}\hat{D}_{t+\ell}.$$

Chapter 5

Cluster Factor GARCH Model

With the review of clustering techniques and multivariate GARCH models finished, the Cluster Factor GARCH Model can then be introduced. The Cluster Factor GARCH model uses factors that are the prototypes found by a clustering algorithm. This novel model draws strength from clustering algorithms' potential in being able to find a small number of representative objects out of a large dataset. These representative objects are designed to be sufficiently dissimilar to each other, while as a collective they should be able to accurately represent the entire dataset. However, in order to do this, a large number of series need to be considered to get an accurate clustering result. This can limit the applicability of the model to markets with a substantial number of assets.

5.1 Model

Consider a set of time series $\{x_{i,t}\}_{i=1}^{M}$ with M being sufficiently large. The next step in the model would be to choose the similarity measure and the clustering algorithm to be used to generate the prototypes. Thus a partitional clustering algorithm is the most natural algorithm to choose. Then, the clustering algorithm should be run to generate k prototypes which will be the factors in the multivariate GARCH model and will be denoted as $f_t = (f_{1,t}, \ldots, f_{k,t})'$. A drawback of this model is that for most partitional clustering algorithms the number of clusters must be chosen a priori or estimated based on some criteria. Then, the set of time series that are to be considered for estimation should be chosen and will be denoted as $r_t = (x_{i_1,t}, \ldots, x_{i_N,t})'$ where $N \leq M$. The Cluster Factor GARCH model can then be modeled as follows

$$r_t = \mu_t + a_t$$
$$a_t = Gf_t + \varepsilon_t$$

which can be seen to be similar to a general factor GARCH model. The model assumptions are the same of those in the general Factor GARCH model. The forecasts of this model will then also be of the same form as the general Factor GARCH model described in Section 4.4.3.

5.1.1 Estimation

This model can be estimated by maximum likelihood where the log-likelihood of the model when the distribution is assumed to be normal is

$$\log L(\theta; r_t) = \sum_{t=1}^{T} \ell_t(\theta)$$
$$\ell_t(\theta) = -\frac{1}{2}\left[N \log 2\pi + \log|G\Sigma_t G' + \Omega| + a_t'(G\Sigma_t G' + \Omega)^{-1} a_t\right]$$

where $\Sigma_t = \text{diag}(\sigma_{1,t}^2, \ldots, \sigma_{k,t}^2)$, $\sigma_{i,t}^2 = \text{Var}[f_{i,t}|\mathcal{F}_t]$, $\Omega = \text{Var}[\varepsilon_t]$, and $\theta = ((\text{Vec } G)', (\text{Vec } \Omega)', \phi)'$ where ϕ is a parameter vector for each of the univariate GARCH parameters for the factors.

However, this likelihood function is hard to directly optimize with an iterative method, so instead the estimation is done with the method detailed in Lin (1992). The outline of the method is done as follows using a standard GARCH(1,1) model for the univariate GARCH model for each of the factors. The first step in this procedure is to estimate the univariate GARCH parameters for each factor to obtain $\hat{\phi}_i = (\hat{\alpha}_{i,0}, \hat{\alpha}_{1,i}, \hat{\beta}_{1,i})'$. Then, the likelihood should be optimized for each individual series to get $\hat{g}_i$ and $\hat{\omega}_{i,i}$ where g_i is the i-th row of G. The log-likelihood function for each series in this stage is

$$\log L(g_i, \omega_{i,i}) = -\frac{1}{2}\sum_{t=1}^{T} \log 2\pi + \log|\hat{h}_{i,t}| + \frac{a_{i,t}^2}{\hat{h}_{i,t}}$$
$$\hat{h}_{i,t} = \omega_{i,i} + \sum_{j=1}^{k} g_{i,j}^2(\hat{\alpha}_{i,1} f_{j,t-1}^2 + \beta_{i,j}\hat{\sigma}_{j,t-1}^2)$$

where $\hat{\sigma}_{j,t}^2$ is the estimated conditional variance for the j-th factor. The equation for $\hat{h}_{i,t}$ is also the only term that will change on the model specification for the univariate GARCH model. If instead the univariate GARCH model was chosen to be an EGARCH(1,1) model, $\hat{h}_{i,t}$ would instead be defined by

$$\hat{h}_{i,t} = \omega_{i,i} + \sum_{j=1}^{k} g_{i,j}^2(\sigma_{j,t}^{2\beta_j} \exp\{\theta z_t + \gamma(|z_t| - \mathbb{E}[|z_t|])\}).$$

Finally, if Ω is assumed to not be diagonal, then the off-diagonal elements of Ω can estimated as follows

$$\hat{\omega}_{i,j} = \frac{1}{T}\sum_{t=1}^{T} a_{i,t}a_{j,t} - \frac{1}{T}\sum_{n=1}^{k} \hat{g}_{n,i}\hat{g}_{n,j}\sum_{t=1}^{T} f_{n,t}^2.$$

Lin (1992) has shown that this method of estimation will lead to consistent estimates of all parameters, however the standard errors would need to be corrected for proper inference.

However, there is another potential method to obtain estimates of the factor loadings matrix G. If the clustering algorithm chosen is a fuzzy clustering algorithm, then the degree of

memberships can be used in place of having to estimate the factor loadings. The factor loadings will then be $g_j = U_{i_j}$ where i_j is the index of asset j in the dataset, and U_i is the i-th row of the membership matrix U from the fuzzy clustering algorithm. When this method for factor loadings is used, the model is then denoted as the U Cluster Factor GARCH model. A problem with this method is that if $k < N$, then the resulting variance matrix will be singular. This will not necessarily be a problem in general for the model as Ω will be non-singular and in most cases the resulting variance matrix will still be non-singular.

Chapter 6

Empirical Application

To gauge the performance of the cluster Factor GARCH model, an application on real stock return data is used. First, a risk management application is done in order to test the specification of the model. Then, the model's forecasting ability is tested by a comparison to other available multivariate GARCH models.

6.1 Data

The data used consists of adjusted closing prices for all stocks currently in the S&P 500 and for intraday price data on a portfolio of nine stocks. The intraday price data is collected at 5 minute intervals so that there are 78 price observations for each full trading day. The data collected spans from 11 January 2000 to 31 January 2021 containing 5031 days and for forecasting purposes, the data is cleaned to remove any partial trading days which leaves the dataset to consist of 4930 observations. Each stock's price is transformed to log returns and multiplied by 100, for numerical purposes, so that for a stock with prices $\{p_t\}_{t=0}^{T}$, the series is transformed to

$$r_t = 100 \log \frac{p_t}{p_{t-1}} \quad \text{for } t = 1, \ldots, T.$$

Each stock also has its mean removed from the series. Then, to run the following analyses, the nine most active stocks over the span of the data in the S&P 500 were chosen where their basic descriptive statistics can be found in Table 6.1 and a visualization of their returns can be seen in Figure 6.1.

6.2 Methodology

6.2.1 Clustering

To estimate the cluster factor GARCH model, a clustering algorithm needs to be run. The first choice that needs to be made is the dissimilarity measure that should be used. To fully use the structure of the data, Dynamic Time Warping (DTW) is used as the clustering measure. In

Name	Symbol	Min	Median	Max	Std. Dev.	Skewness	Kurtosis
Advanced Micro Devices	AMD	-39.20	-0.043	42.02	3.852	-0.182	10.04
Apple	AAPL	-19.86	-0.020	12.91	2.290	-0.045	4.31
AT&T	T	-9.07	0.030	15.08	1.541	0.248	6.83
Bank of America	BAC	-34.22	0.011	30.20	2.905	-0.269	27.31
Cisco	CSCO	-17.68	0.044	14.81	2.187	-0.274	7.50
General Electric	GE	-13.67	-0.013	18.00	2.064	0.214	7.77
Micron Technology	MU	-26.19	0.001	18.46	3.395	-0.312	4.08
Microsoft	MSFT	-17.00	-0.007	17.02	1.811	-0.032	8.28
Wells Fargo	WFC	-27.23	-0.022	28.32	2.438	0.088	25.79

Table 6.1: Stock Names and Descriptive Statistics

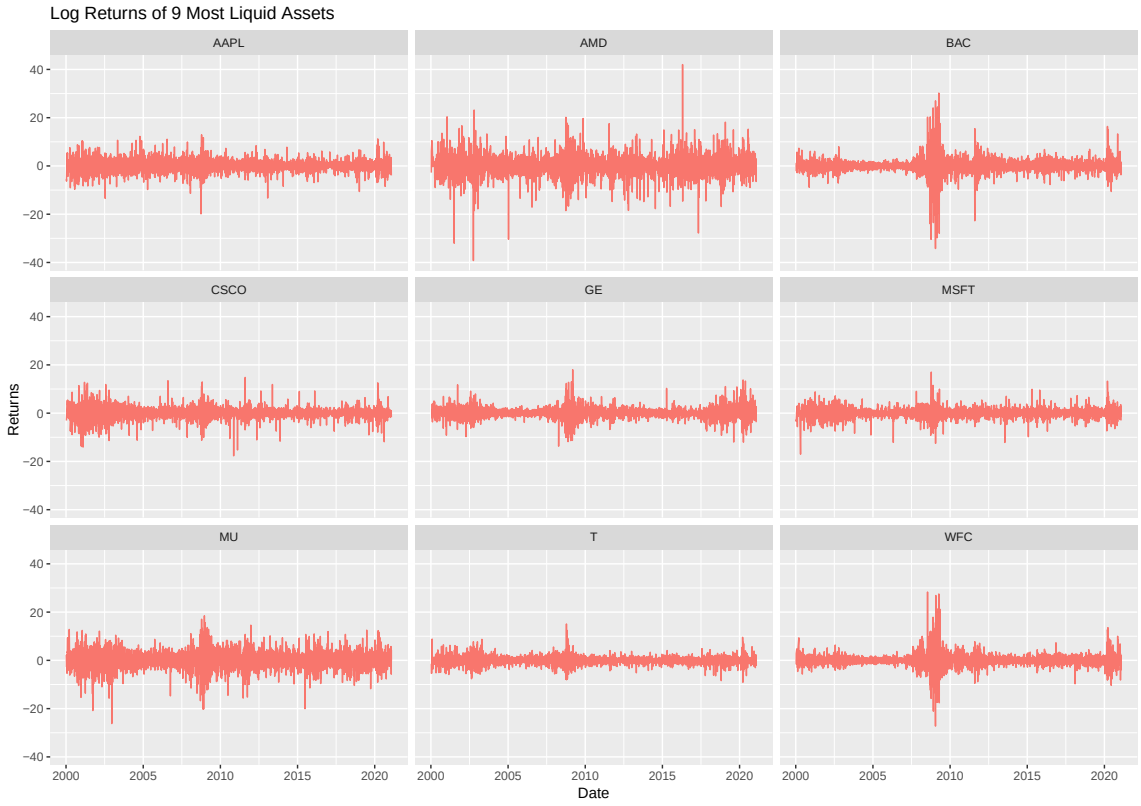

Figure 6.1: Return Series for each Stock

addition to the dissimilarity measure, a representation for each time series needs to be considered. The raw time series will be considered as well as the Piecewise Aggregate Approximation (PAA). The PAA representation is epecially useful, since the DTW algorithm is quadratic in complexity and using PAA can reduce the computational burden. The PAA represenation of each time series is chosen to consist of 1768 observations which is roughly one third of the size of the original dataset. This will decrease the time taken by the DTW dissimilarity calculation by a factor of $\frac{1}{9}$ which can lead to a drastic increase in speed of the calculations of the pairwise dissimilarities. The dissimilarities based on the raw representation and the PAA representation will both be used in the subsequent clustering algorithms. Then, a clustering algorithm needs to be chosen and for ease of calculation, medoid methods are chosen. With k-Medoids, the main two options are the Partition Around Medoids (PAM) algorithm and the Fuzzy c-Medoids (FCMdd) algorithm, both of which will be considered. Both PAM and FCMdd however require the user to specify the number of clusters k before running the algorithm. There are also multiple things that need to be considered when using FCMdd, as the algorithm uses a random starting point and the fuzzifier constant, m, will both impact the final cluster. To account for the specification of k, both algorithms are performed with $k = 2 \ldots, 9$. Then, to account for the impact of different levels of m on FCMdd, the algorithm is performed with $m = 1.01$ and $m = 1.1$ for each level of different cluster. However, accounting for the random start indices is more difficult as it would be computationally infeasible to test out all possible starting indices to determine which provides the best fit. Instead, a greedy selection procedure is used to determine which starting indices will provide the best result. This is done by trying 100 different random starting indices and running the algorithm for only 50 iterations and then the result that provides the lowest cost is chosen and is then run in full.

However, this method of performing a large number of clusters causes it to be difficult to choose which clustering will lead to the best results in the factor model. Instead, each clustering's medoids will be used to estimate the Cluster Factor GARCH model and then the models chosen for further models are chosen based on their Bayesian Information Criterion which for the Cluster Factor GARCH model is

$$BIC = \left(Nk + \frac{1}{2} N(N+1) \right) \log T - 2 \sum_{i=1}^{N} \log L(\hat{g}_i, \hat{\omega}_{i,i})$$

where N is the number of series, k is the number of clusters, T is the number time points, and $\hat{g}_i$ and $\hat{\omega}_{i,i}$ are the estimated maximum likelihood parameters. This procedure is also done for the U Cluster Factor GARCH model, where instead only the results of the FCMdd algorithm are considered. The above BIC formula will then change so that $\hat{g}_i$ will instead be based on the membership matrix. This method is used for both determining the optimal factors for both standarad GARCH specifications and EGARCH specifications.

6.2.2 Portfolio Analysis

Then, to analyze an equally weighted portfolio of the stocks with the Cluster Factor GARCH model, the one day Value at Risk (VaR) with a 5% coverage is tested. The one day VaR at a coverage level $100\alpha\%$, $\text{VaR}_{t+1|t}(\alpha)$ for the returns series r_t is defined by

$$\mathbb{P}[r_{t+1} < \text{VaR}_{t+1|t}(\alpha)] = \alpha$$

where $\text{VaR}_{t+1|t}(\alpha)$ is typically estimated by the information set $\mathcal{F}_t$. After a realization of r_{t+1}, the hit series of the VaR can be defined as

$$I_{t+1}(\alpha) = \begin{cases} 1 & r_{t+1} < \text{VaR}_{t+1|t} \\ 0 & \text{otherwise.} \end{cases}$$

Then, in order to test a model, a VaR series can be estimated and backtested to see if the model is correctly specified. Christoffersen (1998) states that a series of VaR forecasts are only valid if the following hypotheses are satisfied:

- The probability that a return exceeds the VaR forecast is equal to the coverage rate α
- VaR violations observed at two different dates at the same coverage rate should be independently distributed

where the first hypothesis is the unconditional coverage hypothesis and the second hypothesis is the independence hypothesis. The unconditional coverage hypothesis can be stated as follows

$$\mathbb{P}[I_{t+1}(\alpha) = 1] = \alpha.$$

If both the unconditional coverage and independence hypotheses hold, then the conditional coverage hypothesis is said to hold which then implies that

$$\mathbb{E}[I_{t+1}(\alpha) - \alpha | \mathcal{F}_t] = 0.$$

This then implies that the hit series $\{I_{t+1}(\alpha)\}$ is a series of i.i.d. Bernoulli random variables with probability α. To test the hypothesis of conditional coverage, Christoffersen (2004) suggests to calculate the time between the hits so that the durations between hits can be calculated as

$$d_i = t_i - t_{i-1}$$

where t_i is the time of the i-th hit. Since the series $\{I_{t+1}(\alpha)\}$ is Bernoulli distributed with probability α, the duration series, $\{d_i\}$, can then be seen as a series of Geometric random variables. Based on this, Candelon et al. (2011) proposes a General Method of Moments test that is based on the orthonormal polynomials of the geometric distribution. The orthonormal polynomials of the geometric distribution with probability α are given by the following relation

$$M_{j+1}(d, \alpha) = \frac{(1-\alpha)(2j+1) + \alpha(j-d+1)}{(j+1)\sqrt{1-\alpha}} M_j(d, \alpha) + \frac{j}{j+1} M_{j+1}(d, \alpha) \quad \forall d \in \mathbb{N}$$

with $M_0(d, \alpha) = 1$ and $M_{-1}(d, \alpha) = 0$. Then, Candelon et al. (2011) proposes the following moment condition and test statistic for the conditional coverage hypothesis

$$H_{0,CC} : \mathbb{E}[M_j(d_i, \alpha)] = 0 \quad j = 1, \ldots, p$$
$$J_{CC}(p) = \frac{1}{N} \left(\sum_{i=1}^{N} M(d_i, \alpha) \right)' \left(\sum_{i=1}^{N} M(d_i, \alpha) \right) \xrightarrow{d} \chi^2(p)$$

where p is the number of moment conditions considered and $M(d_i, \alpha)$ is a $p \times 1$ vector containing each of the orthonormal polynomials $M_j(d_i, \alpha)$ for $j = 1, \ldots, p$. If the null hypothesis of conditional coverage is rejected, then Candelon et al. (2011) provides two additional tests to test the hypotheses of unconditional coverage and independence. The moment condition and test statistic for the unconditional coverage hypothesis are

$$H_{0,UC} = \mathbb{E}[M_1(d_i, \alpha)] = 0$$
$$J_{UC} = \frac{1}{N} \left(\sum_{i=1}^{N} M_1(d_i, \alpha) \right)^2 \xrightarrow{d} \chi^2(1)$$

and the moment condition and test statistic for the independence hypothesis are

$$H_{0,\text{Ind}} = \mathbb{E}[M_j(d_i, \beta)] = 0$$
$$J_{\text{Ind}}(p) = \frac{1}{N} \left(\sum_{i=1}^{N} M(d_i, \beta) \right)' \left(\sum_{i=1}^{N} M(d_i, \beta) \right) \xrightarrow{d} \chi^2(p-1)$$

where β is the true violation rate that may not be equal to that of the coverage rate α. In order to calculate J_{Ind}, a consistent estimator β needs to be used which can simply be the Maximum Likelihood Estimator of β which is $1/\bar{d}$ where $\bar{d} = 1/N \sum_{i=1}^{N} d_i$.

To implement this test, the VaR of the equally weighted portfolio of the considered stocks needs to be estimated. First, the Cluster Factor GARCH model needs to be estimated, and to do this a fixed forecasting scheme is used. The T observations are first split into an in-sample part with R observations and an out-of-sample part consisting of P observations. First, the Cluster Factor GARCH model is estimated on the first R observations. Then, one day ahead volatility forecasts are calculated using the estimated parameters for the next P observations. The one day ahead forecast of the variance of the portfolio returns, $w'r_t$ where w is a vector of portfolio weights, is calculated as follows

$$\hat{\text{Var}}[w'r_{t+1}|\mathcal{F}_t] = \hat{\text{Var}}[w'a_{t+1}|\mathcal{F}_t] = w'(\hat{G}\hat{\Sigma}_{t+1}\hat{G}' + \hat{\Omega})w.$$

Then, given that a_t is normally distributed, the series of one day VaR of the portfolio can be calculated as

$$\text{VaR}_{t+1|t}(\alpha) = z_\alpha \sqrt{w'(\hat{G}\hat{\Sigma}_{t+1}\hat{G}' + \hat{\Omega})w} \quad \text{for } t = R, \ldots, T-1$$

where z_α is the α quantile of the normal distribution. From $\{\text{VaR}_{t+1|t}\}_{t=1}^{P}$, the hit series $\{I_{t+1}(\alpha)\}$ and the durations, $d_1, \ldots, d_N$, can be calculated. Then, the test statistics $J_{CC}(p)$,

J_{UC}, and J_{Ind} are calculated. In order to be able to properly test the hypothesis, the critical values need to be approximated. Finding the correct critical values is complicated by the presence of estimation risk as noted by Candelon et al. (2011) and Olmo and Escanciano (2011) which implies that the asymptotic critical values might not be sufficient for inference. To accurately calculate the critical values, two resampling methods are used, the subsampling approach of Politis and Romano (1994) and a non-overlapping block bootstrap. Olmo and Escanciano (2011) demonstrates that the block bootstrap will have better size properties when compared to subsampling, but the block bootstrap assumptions may not hold for all data and may lead to incorrect inference.

Subsampling is a resampling method that is more applicable under minimal assumptions when compared to the block bootstrap. Subsampling in a time series context creates $T - b + 1$ subsamples of the original dataset each with length b where the subsamples have the following form

$$(X_i, X_{i+1}, \ldots, X_{i+b-1}) \quad \text{for } i = 1, \ldots, T - b + 1.$$

Then, similar to the original dataset, the subsamples are each split into an in-sample part consisting of R_b observations and an out-of-sample part consisting of P_b observations where $\frac{P_b}{R_b} \approx \frac{P}{R}$ and $P_b + R_b = b$. Then J_{CC} is calculated using each subsample in a similar procedure as for the full dataset to get $J_{CC}^{(i)}$ for $i = 1, \ldots, T - b + 1$. The critical values can then be calculated as $J_{CC}^{[(1-\eta)(T-b+1)]}$ where η is the nominal size of the test and $J_{CC}^{[i]}$ is the i-th order statistic. The same procedure is done for the test statistics J_{UC} and J_{Ind}.

Block bootstrap is a generalization of the bootstrap of Efron (1992) for time series data. The exact procedure implemented is described in Olmo and Escanciano (2011). The block bootstrap will construct a sample of length T of the form $B_1, B_2, \ldots, B_b$ where $Time = bl$ and l is the length of the blocks. The blocks B_i are constructed from l consecutive observations so that $B_i = (X_{1+(i-1)l}, \ldots, X_{il}$ for $i = 1, \ldots, b$. Then, for each bootstrapped sample, the sample is split into an in-sample part of R observations and an out-of-sample part consising of P observations. Then, J_{CC} statistic is calculated for each sample to get $J_{CC}^{(i)}$ for $i = 1, \ldots, B$. The critical values are then calculated can then be calculated similar to the method used for subsampling but now the critical values are of the form $J_{CC}^{[(1-\eta)B]}$. The same procedure is done for the test statistics J_{UC} and J_{Ind}

This procedure to test the viability of the Value at Risk forecasts is applied to test if the specification of the Cluster Factor GARCH model is suitable for this portfolio. The Cluster Factor GARCH model is estimated with standard GARCH univariate factor processes and is considered with estimated factor loadings and with the factor loadings being the membership matrix U. The data is split so that $R = 2500$ and $P = 2430$, so for subsamples $R_b = 508$ and $P_b = 492$. The test is then implemented with $p = 3$ moment conditions, the length of the

subsamples is 1000, and the number of bootstrap datasets is 750 with a block length of $l = 50$. The analysis is done with a significance level of $\eta = 0.05$.

6.2.3 Forecasting

In order to gauge the forecasting performance of the Cluster Factor GARCH model, the Model Confidence Set (MCS) test developed by Hansen et al. (2011b) is used. The Model Confidence Set test is a statistical test that takes in a number of models and returns a set of models where the best performing model is within this set with a confidence level of $1 - \eta$, where η is the significance level of the test. This approach can then be used to determine which set of models is able to best forecast the future outcomes.

A main problem with determining forecasting accuracy in multivariate GARCH models is how to measure the accuracy. This is difficult due to the output of multivariate GARCH models, conditional variance matrices, being unobserved. In order to be able determine the accuracy, a suitable proxy for the conditional variance matrix needs to be used. This proxy is the realized covariation estimate as detailed in both Andersen et al. (2003) and Barndorff-Nielsen and Shephard (2004). In order to calculate the realized covariation estimate of the conditional variance matrix, intraday price data is needed. Let Δ denote the sampling frequency of the intraday data where typical values are 1 minute, 5 minutes, or 30 minutes. Let $p_{i,t}^{(j)}$ denote the logarithm of the price of stock i on day t at time j for $j = 0, \ldots, \lfloor \frac{1}{\Delta} \rfloor$. Then, the log-returns, $r_t^{(j)} = (r_{1,t}^{(j)}, \ldots, r_{N,t}^{(j)})'$ is then calculated as

$$r_t^{(j)} = p_t^{(j)} - p_t^{(j-1)} \quad j = 1, \ldots, \left\lfloor \frac{1}{\Delta} \right\rfloor.$$

The realized covariation for day t, denoted as $\hat{\Sigma}_t^{\Delta}$, is then defined to be

$$\hat{\Sigma}_t^{\Delta} = \sum_{j=1}^{\lfloor \frac{1}{\Delta} \rfloor} r_t^{(j)} r_t^{(j)'}.$$

Barndorff-Nielsen and Shephard (2004) shows that the realized covariation estimator is consistent as $\Delta \to 0$.

With a suitable proxy for the conditional variance matrix determined, the next step is to measure the difference between the proxy and the estimate. This can be done with loss functions. A quadratic loss function for the forecast error can be

$$L_t = \text{Vech}(\hat{H}_t - \hat{\Sigma}_t^{\Delta})' \text{Vech}(\hat{H}_t - \hat{\Sigma}_t^{\Delta}).$$

This particular loss function is the Euclidean distance between $\hat{H}_t$ and $\hat{\Sigma}_t^{\Delta}$ in the space of half vectorized $N \times N$ matrices. Other loss functions could be the Stein loss or the Frobenius distance between $\hat{H}_t$ and $\hat{\Sigma}_t^{\Delta}$.

The MCS test is then based on the differences between the loss values of different models. The initial step is to construct the initial model set $\mathcal{M} = \{\hat{H}_{i,t+k} : i = 1, \ldots, M\}$ where $\hat{H}_{i,t+k}$ is the k day ahead forecast from the i-th model, $t = 1, \ldots, T$, T is the number of forecasts, and M is the number of models considered. The null hypothesis of the test is that all models in $\mathcal{M}$ have equal forecasting performance. Then, the loss values for each model and each day is calculated and denoted as $L_{i,t}$ for $i = 1, \ldots, M$ and $t = 1, \ldots, T-1$. Then, the relative performance values are calculated as $d_{i,j,t} = L_{i,t} - L_{j,t}$. The following loss statistics are then calculated

$$\bar{d}_{i,j} = \frac{1}{T}\sum_{t=1}^{T} d_{i,j,t} \quad \bar{d}_i = \frac{1}{M}\sum_{j\in\mathcal{M}} d_{i,j}$$

where $\mathcal{M}$ is the set of models considered. However, it may be easier to instead calculate the following quantities

$$\bar{L}_i = \frac{1}{T}\sum_{t=1}^{T} L_{i,t} \quad \bar{L} = \frac{1}{M}\sum_{i\in\mathcal{M}} \bar{L}_i \quad \bar{d}_i = \bar{L}_i - \bar{L}.$$

The null hypothesis of the test can now be written as $H_0 : \mathbb{E}[\bar{d}_i] = 0$. From $\bar{d}_i$, t-statistics are calculated for each model as follows

$$t_i = \frac{\bar{d}_i}{\sqrt{\mathrm{Var}[\bar{d}_i]}}.$$

Then, the test statistic is computed to be

$$T_{\mathcal{M}} = \max_{i\in\mathcal{M}} t_i$$

which is the test statistic reported in Hansen et al. (2011b) and differs from the test statistic used in Laurent et al. (2012) where they instead use the average of the square t-statistics. Then, if the null hypothesis is rejected, there is an elimination rule $e_{\mathcal{M}}$ that removes a model from $\mathcal{M}$ and the procedure is repeated with the now smaller set $\mathcal{M}$. The variance of distribution of $\bar{d}_i$ however is not analytically tractable and instead needs to be estimated by a block bootstrap procedure. The exact procedure for this is given by Hansen et al. (2011a). The bootstrap scheme is done via a block format so that the block length l needs to be chosen. Then a bootstrap dataset of losses is created by

$$L^*_{b,i,t} = L_{i,\tau_{b,t}} \quad b = 1, \ldots, B \; i = 1, \ldots, M, t = 1, \ldots, T$$

where B is the number of bootstrap samples and $\tau_{b,t}$ is the index for the b bootstrap sample and time t in that sample. Then, bootstrap sample averages of the loss can be calculated as $\bar{L}^*_{b,i} = \frac{1}{M}\sum_{t=1}^{T} L^*_{b,i,t}$ and the bootstrap error can be calculated as $\zeta^*_{b,i} = \bar{L}^*_{b,i} - \bar{L}_i$. Then, an estimate of $\mathrm{Var}[\bar{d}_i]$ can be calculated via bootstrap as follows

$$\hat{\mathrm{Var}}[\bar{d}_i] = \frac{1}{B}\sum_{b=1}^{B}(\zeta^*_{b,i} - \zeta^*_b)^2, \quad \zeta^*_b = \frac{1}{m}\sum_{i\in\mathcal{M}} \zeta^*_{b,i}.$$

With this the t-statistics can be estimated and the test statistic $T_{\mathcal{M}}$ can be calculated from the estimated t-statistics. Then, in order to determine whether to reject the null hypothesis or not,

the empirical distribution of the test statistic is estimated via bootstrap estimates of the test statistic. The bootstrap estimates of the test statistic are calculated as follows

$$T^*_{b,\mathcal{M}} = \max_{i \in \mathcal{M}} t^*_{b,i} \quad b = 1, \ldots, B$$
$$t^*_{b,i} = \frac{\zeta^*_{b,i} - \zeta^*_b}{\hat{\text{Var}}[\bar{d}_i]}.$$

The bootstrap p-values are then calculated as

$$p_{\mathcal{M}} = \frac{1}{B} \sum_{b=1}^{B} 1\{T_{\mathcal{M}} < T^*_{b,\mathcal{M}}\}$$

where $1\{\}$ is the identity function. The null hypothesis is rejected if the p-value $p_{\mathcal{M}}$ is less than the significance level of the test η. If the null hypothesis is rejected, the worst performing model is removed from $\mathcal{M}$ based on the elimination rule $e_{\mathcal{M}}$. This procedure is repeated until there is no rejection of the null hypothesis. The final resulting model set $\mathcal{M}$ is denoted as $\hat{\mathcal{M}}_{1-\alpha}$ and is known as the $100(1-\eta)\%$ Model Confidence Set.

To implement this to compare the Cluster Factor GARCH models to other available multivariate GARCH models, the initial model confidence set is composed of a scalar BEKK model, the RiskMetrics model, the GOGARCH model, the DCC model, the Cluster Factor GARCH model, and the U Cluster Factor GARCH model. The GOGARCH model is that of Zhang and Chan (2009) and is as implememented by Ghalanos (2019) which also implements the DCC model. For the GOGARCH, DCC, and Cluster Factor GARCH models, the univariate specification is for both the standard GARCH(1,1) model and the EGARCH(1,1) model. The dataset consists of 4891 days of closing prices and is split to generate 2332 forecasts. The models are each estimated based on a rolling procedure so that model parameters are re-estimated every 22 days and only the data are updated for each forecast. The estimation of the models is based on the previous 2559 observations. For each day, three forecasts are made for each model at varying horizons. The forecast horizons considered are 1 day, 5 days, and 20 days. The bootstrap procedure for the Model Confidence Set test is tested with block lengths $l = 2, 4$ and the elimination rule $e_{\mathcal{M}}$ used is

$$e_{\mathcal{M}} = \arg\max_{i \in \mathcal{M}} t_i.$$

The analysis is done with a signficance level of $\eta = 0.25$ similar to that of Laurent et al. (2012).

6.3 Results

The following sections on the results of the empirical applications rely on the chosen Cluster Factor GARCH models. The chosen Cluster Factor GARCH model with standard GARCH factors uses factors found from the PAM algorithm with $k = 8$ medoids and also used the full dataset to calculate its distances. The chosen Cluster Factor GARCH model with EGARCH factors uses factors found from the PAM algorith with $k = 7$ with the full dataset to calculate its

distances. The U Cluster Factor GARCH model that is used with both standard GARCH and EGARCH factors uses factors and membership matrix from the results of the FCMdd algorithm with $c = 2$ medoids, a fuzzifier constant of $m = 1.01$, and the distances were calculated based off of the PAA representation.

6.3.1 Portfolio Analysis

A visualization of the estimated one day 5% Value at Risk of both the Cluster Factor GARCH and the U Cluster Factor GARCH is given in figure 6.2 and descriptive statistics for the set of durations for each model are provided in table 7.1. The result of the backtests are also available in table 6.2.

A complication for the inference of the backtest is that for both resampling methods there are multiple samples where 0 or 1 violations occur in the estimated Value at Risk. This complicates the problem as the test statistics require at least 2 violations to be calculated. The subsampling method lead to 37% of samples having 0 or 1 violations for the Cluster Factor GARCH model and 2% for the U Cluster Factor GARCH model, while for the bootstrap 43% of samples had insufficient violations for the Cluster Factor GARCH model and no samples without insufficient violations for the U Cluster Factor GARCH model. Another potential problem is in the estimation of the durations for the Cluster Factor GARCH model on the full sample, there is potentially an extreme duration calculated with 1009 days between violations. This duration is longer than the block size in the subsampling which may complicate how well the subsampling method is able to approximate the distribution of the test statistic if this duration is allowed under the null hypothesis. This problem need not interfere with the inference in the bootstrap scheme however, as the samples are long enough for similar large durations to occur.

The inference problems can be seen in the difference between the results between the subsampling and the bootstrap for the conditional coverage test of the Value at Risk of the Cluster Factor GARCH model. According to the subsampling critical values, the null hypothesis of conditional coverage should be rejected while under the bootstrap the null hypothesis should not be rejected. The problems in inference for the subsampling approach are further revealed that while the conditional coverage hypothesis is rejected neither unconditional coverage hypothesis nor the independence hypothesis are rejected. If the unconditional hypothesis and the independence hypothesis are both not rejected, then the conditional coverage hypothesis should hold.

If the bootstrap inference for the Cluster Factor GARCH model is to be believed, then the backtests suggest that the Cluster Factor GARCH model is correctly specified for the portfolio as none of the null hypotheses are rejected. However, for the U Cluster Factor GARCH model, the model seems to be misspecified as both the bootstrap and subsampling inference leads to

Model	Test	Value	Subsampling p-value	Bootstrap p-value	Asymptotic p-value
Cluster Factor GARCH	Conditional Coverage	26952	0.000	0.528	0.000
	Unconditional Coverage	0.215	0.973	0.974	0.643
	Independence	91.83	0.650	0.085	0.000
U Cluster Factor GARCH	Conditional Coverage	751.04	0.037	0.000	0.000
	Unconditional Coverage	260.88	0.052	0.000	0.000
	Independence	748.87	0.037	0.000	0.000

Table 6.2: Backtest Test Statistics and p-values

rejection of the null hypothesis of conditional coverage. This may be due to the forecasted conditional variance matrices being singular since the number of factors is less than 9. Thus, using the U Cluster Factor GARCH model may not lead to a proper model for modelling the conditional variance matrix. However, a similar statement cannot be made about the Cluster Factor GARCH model as the inference is unclear.

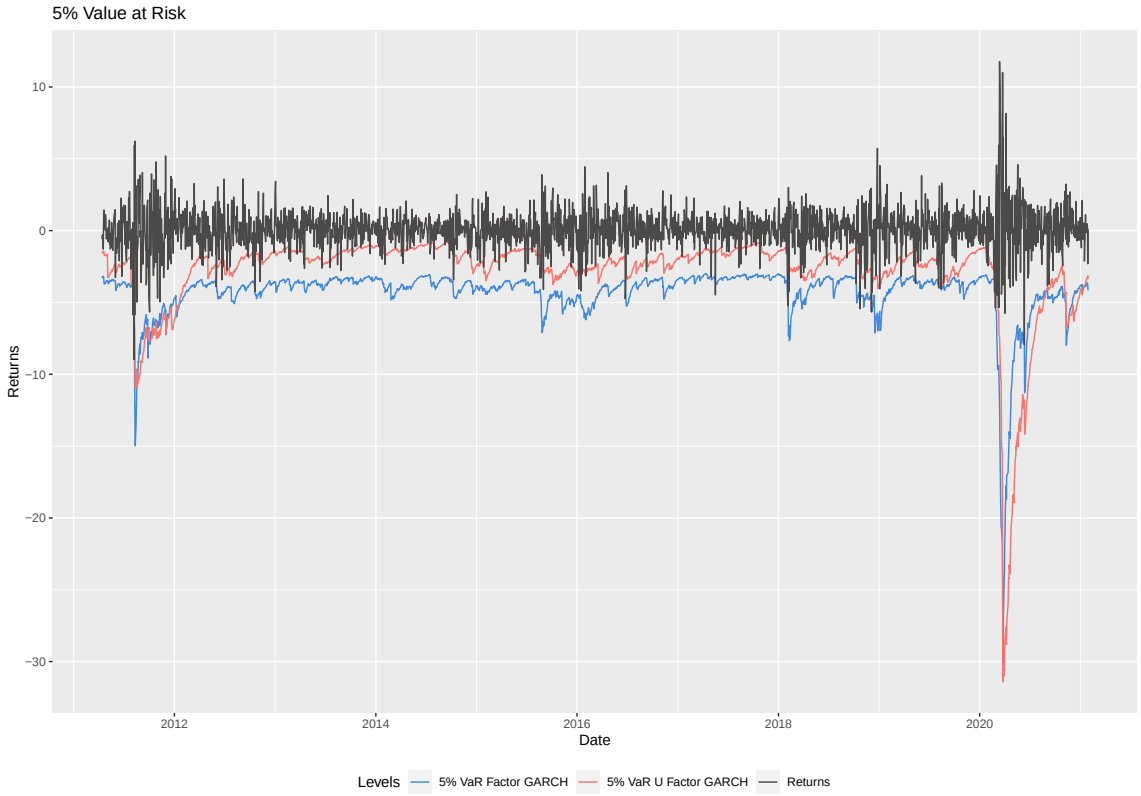

Figure 6.2: 5% Value at Risk for Equally Weighted Portfolio

6.3.2 Forecasting

The Model Confidence Set Results for 1, 5, and 20 day forecasts with a block bootstrap length of $l = 2$ are provided in Tables 6.3, 6.4, and 6.5 and for a block length of $l = 4$, the results are

provided in Tables 7.2, 7.3, and 7.4. A visualization of the 5 day forecasts of the models in the Model Confidence Set can be found in Figure 6.3.

The GOGARCH, DCC, Cluster Factor GARCH, and U Cluster Factor GARCH models for both standard GARCH and EGARCH specifications were present in all of the estimated Model Confidence Sets regardless of forecast horizon or bootstrap block length. Interestingly, the U Cluster Factor GARCH was able to outperform the Cluster Factor GARCH model regardless of horizon, bootstrap block length, and univariate GARCH specification when considering the average loss value. This might suggest that the estimated factor loadings in the Cluster Factor GARCH model are no longer accurate out of the sample they were estimated in. The U Cluster Factor GARCH was also able to outperform the DCC model with standard GARCH specification and the BEKK type models in most circumstances.

With both Cluster Factor GARCH models being in all of the Model Confidence Sets, both models can be seen as having statistically equivalent forecasting abilities as the GOGARCH and DCC models. However, this result only holds when using the Euclidean distance as the loss function. The results could change if instead an asymmetric loss function was used. Indeed Laurent et al. (2012) finds that asymmetric loss functions lead to smaller Model Confidence Sets which could lead to the Cluster Factor GARCH models no longer being equivalent in forecasting performance to GOGARCH and DCC. Another problem in this finding might be that the set of model considered is relatively small, and it could be that with the introduction of more models, the models would no longer be equivalent to the best performing models. However, if a symmetric loss function is chosen and the other models being considered are those considered here, the Cluster Factor GARCH models are valid alternatives to the GOGARCH and DCC models for forecasting applications. This finding however may be out of line with the findings of the last section where the Cluster Factor GARCH models were not found to be correctly specified. If the only requirement for forecasts is their reliability when compared to a realized measure, then the Cluster Factor GARCH models seem to be able to reliably provide accurate forecasts. If instead the model is used for risk management purposes where the model should be correctly specified, the models seem to not be able to adequately model the observed processes.

l = 2

Model	Univariate GARCH	Rank	$\bar{L}_i$	t_i	p-value
GOGARCH	GARCH(1,1)	6	11.89	0.417	0.739
	EGARCH(1,1)	1	7.58	-1.559	1.000
DCC	GARCH(1,1)	7	12.12	0.535	0.684
	EGARCH(1,1)	2	10.41	-0.281	0.955
Cluster Factor GARCH	GARCH(1,1)	8	12.97	0.918	0.491
	EGARCH(1,1)	5	11.69	0.345	0.771
U Cluster Factor GARCH	GARCH(1,1)	4	10.71	-0.122	0.927
	EGARCH(1,1)	3	10.49	-0.212	0.945

Table 6.3: Model Confidence Set Results for 5 day: Rank is based on $\bar{L}_i$

Figure 6.3: 5 day ahead forecasts for each model in the MCS

l=2

Model	Univariate GARCH	Rank	$\bar{L}_i$	t_i	p-value
RiskMetrics		10	12.19	1.110	0.438
BEKK		3	9.66	-0.351	0.980
GOGARCH	GARCH(1,1)	8	11.49	0.608	0.706
	EGARCH(1,1)	1	7.13	-1.553	1
DCC	GARCH(1,1)	6	10.87	0.307	0.837
	EGARCH(1,1)	2	9.06	-0.732	0.997
Cluster Factor GARCH	GARCH(1,1)	9	11.75	0.782	0.614
	EGARCH(1,1)	7	11.01	0.402	0.802
U Cluster Factor GARCH	GARCH(1,1)	4	9.83	-0.250	0.971
	EGARCH(1,1)	5	9.99	-0.154	0.958

Table 6.4: Model Confidence Set Results for 1 day

l = 2

Model	Univariate GARCH	Rank	$\bar{L}_i$	t_i	p-value
GOGARCH	GARCH(1,1)	4	11.79	0.032	0.880
	EGARCH(1,1)	1	8.36	-1.455	1.000
DCC	GARCH(1,1)	5	12.07	0.159	0.855
	EGARCH(1,1)	6	12.44	0.329	0.838
Cluster Factor GARCH	GARCH(1,1)	8	14.07	1.002	0.432
	EGARCH(1,1)	7	13.15	0.631	0.627
U Cluster Factor GARCH	GARCH(1,1)	3	11.06	-0.286	0.954
	EGARCH(1,1)	2	10.80	-0.404	0.968

Table 6.5: Model Confidence Set Results for 20 days: Rank is based on $\bar{L}_i$

Chapter 7

Conclusion

This paper has introduced some of the major topics in both clustering analysis and multivariate GARCH models. A specific analysis of time series clustering allows for some of the peculiarities in time series data to be exploited for further analysis. This all then lead to the formation of the new factor GARCH model, the Cluster Factor GARCH model.The Cluster Factor GARCH model was then applied to two empirical applications to judge its performance. In the first application the Cluster Factor GARCH model was used to model the 5% Value at Risk of an equally weighted portfolio consisting of the most active stocks in the S&P 500. A backtest of the Value at Risk showed that the model may not be correctly specified when using estimated factor loadings matrix and was not correctly specified when using the membership matrix as factor loadings. The second application was in comparing its forecasting performance relative to other common multivariate GARCH models. The Cluster Factor GARCH model was shown to have statistically equivalent forecasting performance to the other common multivariate GARCH models.

Given the breadth of research in cluster analysis, more research can be done on its application to multivariate GARCH models. Other clustering algorithms could be used for generating prototypes such as k-Means or hierarchical methods. A robustness check on the choice of dissimilarity measure could be done to see whether similar results could be achieved when using Euclidean distance or correlation as dissimilarity or similarity measures compared to Dynamic Time Warping. Another choice that could lead to further performance improvements is in the estimation procedure. Santos and Moura (2014) provides an interesting estimation procedure for factor GARCH models with observed factors. This procedure could be used for the Cluster Factor Model to allow for dynamic factor loadings which was not considered in this paper.

Bibliography

Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. (2015). Time-series clustering–a decade review. *Information Systems*, 53:16–38.

Alexander, C. (2001). Orthogonal garch. *Mastering risk*, pages 21–38.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, 71(2):579–625.

Barndorff-Nielsen, O. E. and Shephard, N. (2004). Econometric analysis of realized covariation: High frequency based covariance, regression, and correlation in financial economics. *Econometrica*, 72(3):885–925.

Bauwens, L., Laurent, S., and Rombouts, J. V. K. (2006). Multivariate garch models: a survey. *Journal of applied econometrics (Chichester, England)*, 21(1):79–109.

Begusic, S. and Kostanjcar, Z. (2020). Cluster-specific latent factor estimation in high-dimensional financial time series. *IEEE access*, 8:164365–164379.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327.

Bollerslev, T. (1990). Modelling the coherence in short-run nominal exchange rates: A multivariate generalized arch model. *The review of economics and statistics*, 72(3):498–505.

Bollerslev, T., Engle, R. F., and Wooldridge, J. M. (1988). A capital asset pricing model with time-varying covariances. *The Journal of political economy*, 96(1):116–131.

Boswijk, H. P. and van der Weide, R. (2011). Method of moments estimation of go-garch models. *Journal of econometrics*, 163(1):118–126.

Candelon, B., Colletaz, G., Hurlin, C., and Tokpavi, S. (2011). Backtesting value-at-risk: A gmm duration-based test. *Journal of financial econometrics*, 9(2):314–343.

Christoffersen, P. F. (1998). Evaluating interval forecasts. *International economic review (Philadelphia)*, 39(4):841–862.

Christoffersen, P. F. (2004). Backtesting value-at-risk: A duration-based approach. *Journal of financial econometrics*, 2(1):84–108.

Duarte, F. G. and De Castro, L. N. (2020). A framework to perform asset allocation based on partitional clustering. *IEEE access*, 8:110775–110788.

Efron, B. (1992). Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer.

Engle, R. (2002). Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of business & economic statistics*, 20(3):339–350.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflations. *Econometrica*, 50(4):987–.

Engle, R. F. and Kroner, K. F. (1995). Multivariate simultaneous generalized arch. *Econometric theory*, 11(1):122–150.

Ghalanos, A. (2019). *rmgarch: Multivariate GARCH models.* R package version 1.3-7.

Glosten, L. R., Jagannathan, R., and Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *The Journal of finance (New York)*, 48(5):1779–1801.

Hansen, P. R. (2005). A test for superior predictive ability. *Journal of business & economic statistics*, 23(4):365–380.

Hansen, P. R., Lunde, A., and Nason, J. M. (2011a). Appendix: Model confidence sets. *Econometrica Supplemental Material*, 79(2).

Hansen, P. R., Lunde, A., and Nason, J. M. (2011b). The model confidence set. *Econometrica*, 79(2):453–497.

Huang, J., Chai, J., and Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of business research in China*, 14(1).

Izakian, H., Pedrycz, W., and Jamal, I. (2015). Fuzzy clustering of time series data using dynamic time warping distance. *Engineering Applications of Artificial Intelligence*, 39:235–244.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

Keogh, E. and Pazzani, M. (2000a). Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '00, pages 285–289. ACM.

Keogh, E. J. and Pazzani, M. J. (2000b). A simple dimensionality reduction technique for fast similarity search in large time series databases. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 122–133. Springer.

Krishnapuram, R., Joshi, A., Nasraoui, O., and Yi, L. (2001). Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE transactions on Fuzzy Systems*, 9(4):595–607.

Laurent, S., Rombouts, J. V. K., and Violante, F. (2012). On the forecasting accuracy of multivariate garch models. *Journal of applied econometrics (Chichester, England)*, 27(6):934–955.

Lin, W.-L. (1992). Alternative estimators for factor garch models-a monte carlo comparison. *Journal of applied econometrics (Chichester, England)*, 7(3):259–279.

Morgan, J. (1996). Riskmetrics—technical document. *Morgan Guaranty Trust Company of New York: New York*, 51:54.

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, 59(2):347–370.

Olmo, J. and Escanciano, J. C. (2011). Robust backtesting tests for value-at-risk models. *Journal of financial econometrics*, 9(1):132–161.

Politis, D. N. and Romano, J. P. (1994). Large sample confidence regions based on subsamples under minimal assumptions. *The Annals of statistics*, 22(4):2031–2050.

Sakoe, H. (1971). Dynamic-programming approach to continuous speech recognition. In *1971 Proc. the International Congress of Acoustics, Budapest.*

Santos, A. A. and Moura, G. V. (2014). Dynamic factor multivariate garch model. *Computational statistics & data analysis*, 76:606–617.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., and Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing (Amsterdam)*, 267:664–681.

Sheppard, K. and Engle, R. F. (2001). Theoretical and empirical properties of dynamic conditional correlation multivariate garch.

Silvennoinen, A. and Teräsvirta, T. (2009). Multivariate garch models. In *Handbook of Financial Time Series*, pages 201–229. Springer Berlin Heidelberg, Berlin, Heidelberg.

Tola, V., Lillo, F., Gallegati, M., and Mantegna, R. N. (2008). Cluster analysis for portfolio optimization. *Journal of economic dynamics & control*, 32(1):235–258.

van der Weide, R. (2002). Go-garch: a multivariate generalized orthogonal garch model. *Journal of applied econometrics (Chichester, England)*, 17(5):549–564.

Verma, A., Buonocore, R. J., and Di Matteo, T. (2019). A cluster driven log-volatility factor model: a deepening on the source of the volatility clustering. *Quantitative finance*, 19(6):981–996.

West, K. D. (2006). Forecast evaluation. *Handbook of economic forecasting*, 1:99–134.

Zhang, K. and Chan, L. (2009). Efficient factor garch models and factor-dcc models. *Quantitative finance*, 9(1):71–91.

Appendix

Model	N	Mean	Min	Median	Max	St. Dev
Cluster Factor GARCH	16	146.33	2	69	1009	250.97
U Cluster Factor GARCH	115	19.04	1	10	162	25.26

Table 7.1: Descriptive Statistics for Durations of 5% Value at Risk Violations

l=4

Model	Univariate GARCH	Rank	$\bar{L}_i$	t_i	p-value
RiskMetrics		9	14.47	1.145	0.369
GOGARCH	GARCH(1,1)	6	11.89	0.186	0.848
	EGARCH(1,1)	1	7.58	-1.328	1.000
DCC	GARCH(1,1)	7	12.12	0.272	0.812
	EGARCH(1,1)	2	10.41	-0.369	0.979
Cluster Factor GARCH	GARCH(1,1)	8	12.97	0.554	0.676
	EGARCH(1,1)	5	11.69	0.118	0.873
U Cluster Factor GARCH	GARCH(1,1)	4	10.71	-0.233	0.962
	EGARCH(1,1)	3	10.49	-0.302	0.971

Table 7.2: Model Confidence Set Results for 5 day Forecasts: Rank is based on $\bar{L}_i$

l=4

Model	Univariate GARCH	Rank	$\bar{L}_i$	t_i	p-value
RiskMetrics		10	12.19	0.894	0.540
BEKK		3	9.66	-0.272	0.971
GOGARCH	GARCH(1,1)	8	11.49	0.487	0.747
	EGARCH(1,1)	1	7.13	-1.264	1.000
DCC	GARCH(1,1)	6	10.87	0.247	0.851
	EGARCH(1,1)	2	9.06	-0.586	0.994
Cluster Factor GARCH	GARCH(1,1)	9	11.75	0.608	0.691
	EGARCH(1,1)	7	11.01	0.338	0.813
U Cluster Factor GARCH	GARCH(1,1)	4	9.83	-0.202	0.961
	EGARCH(1,1)	5	9.99	-0.124	0.951

Table 7.3: Model Confidence Set Results for 1 day Forecasts

l=4

Model	Univariate GARCH	Rank	$\bar{L}_i$	t_i	p-value
GOGARCH	GARCH(1,1)	4	11.79	0.024	0.892
	EGARCH(1,1)	1	8.36	-1.138	1
DCC	GARCH(1,1)	5	12.07	0.251	0.855
	EGARCH(1,1)	6	12.44	0.123	0.804
Cluster Factor GARCH	GARCH(1,1)	8	14.06	0.772	0.537
	EGARCH(1,1)	7	13.15	0.496	0.683
U Cluster Factor GARCH	GARCH(1,1)	3	11.06	-0.217	0.956
	EGARCH(1,1)	2	10.80	-0.304	0.968

Table 7.4: Model Confidence Set Results for 20 day Forecasts