Imperial College London

Imperial College London

Department of Mathematics

Detecting multivariate market regimes via clustering algorithms

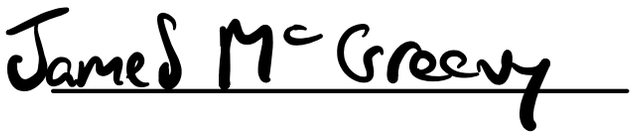
Author: James Mc Greevy (CID: 01075416)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2022-2023

Declaration

The work contained in this thesis is my own work unless otherwise stated.



Acknowledgements

This thesis would not have been possible without the support of many people throughout the academic year. In particular, I would like to thank my colleagues and supervisors Aitor Muguruza Gonzalez, Zacharia Issa, and Jonathan Chan at Kaiju Capital Management, as well as my academic supervisor Cris Salvi for their invaluable support, knowledge and contributions.

I would also like to thank my family, whose love and support throughout my education and career has never wavered, and without whom I would be nothing. I am thankful for my friends at Imperial, Bath, and beyond, for their guidance, assistance, and encouragement while down in the trenches of our numerous coursework assignments and exams in both this and previous academic years. Finally, I want to extend a special thank you to my girlfriend, Jennifer, for her enduring patience, love, and unwavering support, without which the completion of this thesis would not have been possible.

Abstract

In this thesis we study the joint dynamics of multivariate time series using an unsupervised learning technique. We present a novel non-parametric market regime detection method for multidimensional data. The detection procedure is based on a k-means clustering algorithm which makes use of distances between distributions to discriminate between different market regimes. In particular, we empirically investigate the performance of the algorithm endowed with either Wasserstein distances or Maximum Mean Discrepancies. We suggest a two-step approach to clustering multivariate data using this new method and the WK-means algorithm presented in [1] which we show to be effective on both synthetic and real world data. We demonstrate how our new approach can be used to obtain approximations to the mean, variance and correlation between two assets at a given point in time. We further show that these values can be used in the context of Modern Portfolio Theory to form profitable trading strategies when using two assets, in effect pairs-trading.

Contents

1 Introduction **8**
1.1 Clustering with multivariate time series 8
1.2 Thesis outline and contributions 10

2 Multivariate data and distributions **11**

3 Distance metrics and clustering algorithms **13**
3.1 Background 13
3.2 The k-means clustering algorithm 15
3.3 Maximum Mean Discrepancy 17
3.4 p-Wasserstein distance 17
3.5 The WK-means algorithm 18

4 The WK-means and MMDK-means algorithms **21**
4.1 The uni-dimensional WK-means algorithm 21
4.2 The d-dimensional WK-means algorithm 23
4.2.1 Background 23
4.2.2 Approach 24
4.3 The d-dimensional MMDK-means algorithm 25
4.3.1 Approach 25

5 Synthetic data experiments **27**
5.1 Geometric Brownian motion 30
5.1.1 Fixed correlation regime 31
5.1.2 Fixed mean-variance regime 36
5.1.3 Free correlation and mean-variance regime 39
5.2 Merton jump diffusion process 41
5.2.1 Fixed correlation regime 42
5.2.2 Fixed mean-variance regime 46
5.2.3 Free correlation and mean-variance regime 50
5.3 Speed 52
5.4 Summary 53

6 Real data experiments **54**
6.1 Synthetic data experiments revisited 56
6.2 Real data 61
6.2.1 AAPL - AMZN 61
6.2.2 AVB - ESS 65
6.2.3 SJM - PAYC 68
6.3 Selection of hyperparameters 71

7 Trading strategies **75**
7.1 Modern Portfolio Theory 75
7.2 The Magic Genie strategy 76
7.3 Cluster Based trading strategy 79

8 Conclusion **81**

A Technical Proofs and further results **82**
A.1 Appendix 1 82
A.1.1 The k-means clustering algorithm 82
A.1.2 Maximum Mean Discrepancy 82
A.1.3 p-Wasserstein distance 83
A.2 Appendix 2 84
A.3 Appendix 3 85
A.3.1 Geometric Brownian motion 85
A.3.2 Correlation of Merton jump diffusion processes 86
A.3.3 Merton jump diffusion process 87
A.4 Appendix 4 89

Bibliography **94**

List of Figures

2.1 Matrix form of a $d \times n$ multivariate time series. 12

4.1 Barycentre computation flow diagram 25

5.1 Matrix form of a $2 \times h_1$ empirical distribution μ. 28
5.2 Example diagram of the segmentation of returns and the subsequent association of returns with a cluster color. 29
5.3 GBM synthetic price paths with $\rho = 1$, and their associated log-returns. 32
5.4 GBM, $\rho = 1$, example mean-variance, correlation and price series plots. 33
5.5 GBM, $\rho = 0$, example mean-variance, correlation and price series plots. 34
5.6 GBM price paths with four regimes and $\rho = 0$, example plots. 35
5.7 GBM with correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, example mean-variance, correlation and price series plots. 37
5.8 GBM with correlation regimes, $\rho_0 = 0.5$ and $\rho_1 = 1$, example mean-variance, correlation and price series plots. 38
5.9 GBM price paths with three market and correlation regimes that do not overlap segmented by cluster association, $\rho_0 = 0$ and $\rho_1 = 1$, example plots. 40
5.10 GBM price paths with four mean-variance and correlation regimes segmented by cluster association, $\rho_0 = 0$ and $\rho_1 = 1$, example plots. 41
5.11 MJD synthetic price paths with $\rho = 1$, and their associated log-returns. 42
5.12 MJD, $\rho = 1$, example mean-variance, correlation and price series plots. 43
5.13 MJD, $\rho = 0$, example mean-variance, correlation and price series plots. 45
5.14 MJD price paths with four regimes and $\rho = 0$, example plots. 46
5.15 MJD with correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, example mean-variance, correlation and price series plots. 47
5.16 MJD with correlation regimes, $\rho_0 = 0.5$ and $\rho_1 = 1$, example mean-variance, correlation and price series plots. 49
5.17 MJD price paths with mean-variance and correlation regimes that do not overlap segmented by cluster association, $\rho_0 = 0$ and $\rho_1 = 1$, example plots. 50
5.18 MJD price paths with mean-variance and correlation regimes segmented by cluster association, $\rho_0 = 0$ and $\rho_1 = 1$, example plots. 51

6.1 Uni-d 1-WK-means applied to assets 1 and 2 clustering and centroid plots. 57
6.2 Geometric Brownian motion assets 1 and 2 correlation and price series segmented by cluster association. 59
6.3 Merton jump diffusion assets 1 and 2 correlation and price series segmented by cluster association. 60
6.4 Price series and rolling correlation plots of AAPL and AMZN. 62
6.5 Uni-d 1-WK-means applied to AAPL and AMZN mean-variance clustering plots. . 62
6.6 Empirical measures correlation plots of AAPL and AMZN for 2 clusters. 63
6.7 Centroid, cluster and rolling correlation plots of AAPL and AMZN for 2 clusters. . 64
6.8 Price series and rolling correlation plots of AVB and ESS. 65
6.9 Uni-d 1-WK-means applied to AVB and ESS mean-variance clustering plots. . . . 66
6.10 Empirical measures correlation plots of AVB and ESS for 2 clusters. 66
6.11 Centroid, cluster and rolling correlation plots of AVB and ESS for 2 clusters. . . . 67
6.12 Price series and rolling correlation plots of SJM and PAYC. 68

6.13 Uni-d 1-WK-means applied to SJM and PAYC mean-variance clustering plots. . . 69
6.14 Empirical measures correlation plots of SJM and PAYC for 2 clusters. 69
6.15 Centroid, cluster and rolling correlation plots of SJM and PAYC for 2 clusters. . . 70
6.16 Correlation plots of AAPL and AMZN for 2 clusters, $(h_1, h_2) = (140, 133)$. 72
6.17 Correlation plots of AVB and ESS for 2 clusters, $(h_1, h_2) = (210, 203)$. 73
6.18 Correlation plots of SJM and PAYC for 2 clusters, $(h_1, h_2) = (140, 133)$. 74

7.1 Magic Genie and Rolling Average portfolios. 78
7.2 Example diagram of information used to trade with the Cluster Based (CB) strategy. 79
7.3 Cluster Based strategy results. 80

List of Tables

5.1 Possible combinations of joint regimes (JR). 30
5.2 Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = 1$, $n = 50$ runs. 32
5.3 Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = 0$, $n = 50$ runs. 33
5.4 Accuracy scores with 95% CI, GBM synthetic path with four different mean-variance regimes and fixed $\rho = 0$, $n = 50$ runs. 35
5.5 Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = 1$, $n = 50$ runs. 36
5.6 Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 0.5$, $\rho_1 = 1$, $n = 50$ runs. 36
5.7 Accuracy scores with 95% CI, GBM synthetic path, $k = 3$, with correlation and mean-variance regimes, $n = 50$ runs. 39
5.8 Accuracy scores with 95% CI, GBM synthetic path, $k = 3$, with correlation (JR_1) and mean-variance (JR_2) regimes, $n = 50$ runs. 39
5.9 Accuracy scores with 95% CI, GBM synthetic path, $k = 4$, with correlation and mean-variance regimes, $n = 50$ runs. 40
5.10 Accuracy scores with 95% CI, GBM synthetic path, $k = 3$, with correlation (JR_1), mean-variance (JR_2) and joint correlation-mean-variance (JR_3) regimes, $n = 50$ runs. 40
5.11 Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and $\rho = 1$, $n = 50$ runs. 42
5.12 Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and $\rho = 0$, $n = 50$ runs. 44
5.13 Accuracy scores with 95% CI, MJD synthetic path with four different mean-variance regimes and $\rho = 0$, $n = 50$ runs. 44
5.14 Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = 1$, $n = 50$ runs. 46
5.15 Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 0.5$, $\rho_1 = 1$, $n = 50$ runs. 48
5.16 Accuracy scores with 95% CI, MJD synthetic path, $k = 3$, with correlation and mean-variance regimes, $n = 50$ runs. 50
5.17 Accuracy scores with 95% CI, MJD synthetic path, $k = 3$, with correlation (JR_1) and mean-variance (JR_2) regimes, $n = 50$ runs. 50
5.18 Accuracy scores with 95% CI, MJD synthetic path, $k = 4$, with correlation and mean-variance regimes, $n = 50$ runs. 51
5.19 Accuracy scores with 95% CI, MJD synthetic path, $k = 3$, with correlation (JR_1), mean-variance (JR_2) and joint correlation-mean-variance (JR_3) regimes, $n = 50$ runs. 51
5.20 Distance matrix computation time with 95% CI for d assets, $T = 5$. 52
5.21 Computation time with 95% CI for computing one row of the distance matrix ($[\mathcal{W}_{ij}]_{1 \leq j \leq M}$), and for the entire distance matrix W, for T years of data and over 50 runs. 52
5.22 Computation time with 95% CI for *post*-distance matrix computation for n runs of the 2-d WK-means algorithm, for T years of data, and where $k = 2$ computed over 50 runs. 53
5.23 Total computation time with 95% CI for k clusters and n runs, $T = 20$, $d = 2$. . . . 53

6.1 Accuracy scores of uni-d 1-WK-means applied to GBM synthetic paths with mean-variance and correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, $n = 50$ runs. 56
6.2 Accuracy scores of 2-d WK-means and 2-d MMDK-means applied to GBM synthetic paths with mean-variance and correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, $n = 50$ runs. 56
6.3 Accuracy scores of uni-d 1-WK-means applied to MJD synthetic paths with mean-variance and correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, $n = 50$ runs. 58
6.4 Accuracy scores of 2-d WK-means and 2-d MMDK-means applied to MJD synthetic paths with mean-variance and correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, $n = 50$ runs. 58
6.5 Average percentage occurrence of the most common cluster designation for each empirical measure, $n = 200$ runs. 71

7.1 Magic Genie strategy cumulative returns for $\mu_t = 0.05\%, 0.10\%, 0.15\%$. 77
7.2 Magic Genie (MG) vs Rolling Average (RA) strategy statistics for AAPL-AMZN. . 77
7.3 Magic Genie (MG) vs Rolling Average (RA) strategy statistics for AVB-ESS. . . . 77
7.4 Magic Genie (MG) vs Rolling Average (RA) strategy statistics for SJM-PAYC. . . 77

A.1 Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = 0.5$, $n = 50$ runs. 85
A.2 Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = -0.5$, $n = 50$ runs. 85
A.3 Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = -1$, $n = 50$ runs. 86
A.4 Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 1$, $\rho_1 = 0$, $n = 50$ runs. 86
A.5 Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = -1$, $n = 50$ runs. 86
A.6 Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = -1$, $\rho_1 = 0$, $n = 50$ runs. 86
A.7 Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = -0.5$, $\rho_1 = -1$, $n = 50$ runs. 86
A.8 Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = 0.5$, $n = 50$ runs. 86
A.9 Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and fixed $\rho = 0.5$, $n = 50$ runs. 87
A.10 Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and fixed $\rho = -0.5$, $n = 50$ runs. 88
A.11 Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and fixed $\rho = -1$, $n = 50$ runs. 88
A.12 Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 1$, $\rho_1 = 0$, $n = 50$ runs. 88
A.13 Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = -1$, $n = 50$ runs. 88
A.14 Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = -1$, $\rho_1 = 0$, $n = 50$ runs. 88
A.15 Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = -0.5$, $\rho_1 = -1$, $n = 50$ runs. 88
A.16 Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = 0.5$, $n = 50$ runs. 88

Chapter 1

Introduction

1.1 Clustering with multivariate time series

Clustering is a well-known and studied field in unsupervised learning. The goal of a clustering algorithm is to discover patterns and relationships between the input data that are currently unknown. Datum grouped into a particular cluster should have more similarity with data within the same group than with data in another group according to some metric [2]. Naturally, clustering algorithms are particularly useful when we can not immediately categorise data into groups via other means or when we are approaching a new data set with unknown characteristics. Clustering techniques are thus often employed with new time series data such as those of a financial origin. In the field of time series clustering, working with a single time series, referred to as univariate data, is often simpler than working with multiple time series, referred to as multivariate time series, in part due to lower time and memory complexity. Indeed, there is a wide body of research on clustering univariate time series [3] and a variety of clustering methods for time series data have been studied and employed, such as those summarised in [4].

Clustering multivariate time series is a particularly interesting and difficult challenge. When applying clustering algorithms to multivariate data, we open up new areas of study. For example, suppose we wanted to form a series of risk diversified investment portfolios from the Russell 1000 Index. Forming a time series from the returns for each stock, we might then employ a clustering algorithm such as the celebrated k-means algorithm [5] in order to form our portfolios as in [6]. Often such an approach is accompanied by a dimension reduction technique such as Principal Component Analysis (PCA) [7]. If we have M univariate time series of length N then we may store our multivariate time series in an $M \times N$ matrix. PCA can then be used to transform this matrix into a new coordinate system and thus leave us with a smaller $p \times p$ matrix formed from the first p principal components. In reducing the size of our input matrix we benefit from increased speed and we may also hone in on particularly important features of a time series, but we do so at the expense of a loss of information from those components which have been discarded. A variety of multivariate clustering methods are built using PCA and a well-known clustering technique [8] [9] [10].

Of course, instead of clustering each stock into clusters based on the entire multivariate time series, we might cluster *segments* of the multivariate time series. Such clusters might then represent certain periods where the joint dynamics or distribution of the multivariate time series have changed, perhaps due to a *market regime* change from a *bull* to a *bear* market and vice versa. This can be a particularly interesting area of study that can inform both trading strategies and risk management.

The idea of financial markets and time series being composed of a series of higher-return and-lower return cycles, coined bull and bear cycles respectively, has existed for centuries [11]. A formal and universal definition of these terms has not been forthcoming however. At times, some have defined a bull market as being all non-negative returns with all others indicating a bear market [12]. Others have defined the bull (bear) cycle as 'periods of generally increasing (decreasing)

market prices' [13]. In spite of not having any formal definition, these terms are now ubiquitous within finance thanks to their simplicity and the ease with which they can be understood.

The idea of bull and bear markets however is a simplification of the general philosophy of market regimes, where segments of returns, univariate or multivariate, are split into different groups or regimes each characterised by a different underlying distribution [14]. Thus one need not be limited to just two market regimes. Indeed, we might instead divide market returns into a three state model: a bull regime of particularly high returns, a bear regime of particularly low returns and a 'normal' regime of somewhat low, positive returns in the middle [15]. We might otherwise subdivide the bull and bear markets into two states of bull correction and bull, and bear and bear rally respectively as in [16], [17].

The method through which these regimes are determined and classified is referred to as the *market regime clustering problem* (MRCP) and has been the subject of rigorous study. This problem may also be viewed as a subset of studies related to *change point detection*. In such studies, one works to identify change points in financial markets, points at which the underlying regime or model driving the market returns is assumed to have changed [18] and where the number of such change points may be known or unknown [19]. Certain studies work specifically on determining anomalous datum using outlier detection methods [20] [21].

One of the most popular regime detection methods is the Hidden Markov Model [22]. In such a model, we would have an observed process, X, and an unobserved process, S. The observed process X may be the current return of a stock we are interested in while S would be the Markovian latent state or regime that it is currently following. We say that X is observed as we may actively monitor its evolution whereas S is unobserved as we have no method of directly monitoring it. Instead, we use observations of X to approximate the current state S [23]. One drawback of this approach is that it is not model free - we must model the latent state variable, often using a Gaussian distribution [24]. Other approaches for determining regimes include directional change [25] and Bayesian techniques [26].

Classic unsupervised learning techniques have also been deployed more recently on the market regime clustering problem. The fuzzy c-means algorithm, a close relative of the k-means clustering algorithm, has been used to identify time series clusters in stock and sector data as in [27]. These clusters were then used to train single non-linear regime models in order to forecast future stock prices. A more distinctive approach comes from clustering not the time series data itself, but the empirical measures to which each segment of the time series can be associated [28]. Indeed, this will be the approach utilised in this paper which we will endeavour to explain more clearly in the coming chapters.

Many of the techniques discussed may theoretically be applied to both univariate and multivariate time series. They are, however, often beguiled by problems concerning the *curse of dimensionality*, and thus efficiency, in the multivariate case. One must also consider that the underlying distribution characterising a regime is a joint distribution in the case of multivariate time series. In the univariate case, these regimes can be identified by their mean and variance [15]. When working with multivariate time series, other characteristics, such as the correlation between individual univariate series, also play a role. Clustering based on the cross-correlation between time series has been studied before and we refer the reader to the examples listed for further discussion [29] [30] [31].

Why might one want to cluster segments of multivariate time series? Such clusterings might help to inform one's portfolio formation and optimisation [6] or a pairs trading strategy. The idea behind pairs trading is informed by the natural intuition that the returns of related financial instruments should exhibit some form of relationship. The nature of the financial instruments' relationship might be something fundamental. In his book, 'Quantitative trading: how to build your own algorithmic trading business' author Ernest P. Chan creates a pairs trading strategy using the gold ETF GLD and the gold miners ETF GLDX [32, Example 3.6, page 55]. The assumption here is that each price series should move in tandem given their natural relationship. Stocks considered to be in the same economic sector are often used in such trading strategies due to sector wide trends leading to similarities in their price movements. In general, the goal of a pairs trading strategy is to identify a pair of related financial assets and, subsequently, to go long/short in each asset in

order to exploit this relationship and realise returns.

How one goes about forming such a strategy has been studied for years. A good summary of the main statistical techniques used to identify potential pairings is provided in [33]. The study is conducted using financial stocks from the S&P 500 and the techniques include correlation, minimum squared distance, stochastic, stochastic differential residual and co-integration. The authors find that trading strategies based on the minimum squared distance and co-integration approaches generate residual series with better properties. An alternative approach is provided by [34] wherein several unsupervised clustering methods are used to group related stocks together not solely using their price series but also using similar characteristics. These characteristics include momentum factors and firm characteristics led out by [35]. The paper finds statistically significant monthly alpha using this method that is robust but decreases when clusters are formed solely on the basis of price information. The authors surmise that firm characteristics play a non-trivial role in finding potential pairs. Our paper also utilises an unsupervised method in the adapted k-means clustering algorithm but we use these clusters not to group similar stocks but instead, to group the returns of a predetermined pair of stocks.

1.2 Thesis outline and contributions

This thesis expands upon the work of Blanka Horvath, Zacharia Issa, and Aitor Muguruza in [1]. In their paper, they use an unsupervised learning algorithm to effectively cluster one-dimensional financial time series into market regimes with a high degree of success. This algorithm is a variant of the classic k-means algorithm and uses the p-Wasserstein distance as its distance metric. The algorithm is thus dubbed the WK-means algorithm.

Our main contributions are as follows. We create a novel, model-free algorithm that extends the WK-means algorithm such that it may work with data of dimension d for $d > 1$, before showing that this algorithm is effective in detecting, offline, times of market regime change and, in particular, changes in the correlation structure between two assets. We also show that this new algorithm is mutable and can be used with the Maximum Mean Discrepancy instead of the p-Wasserstein distance. We then use our findings in conjunction with [1] to develop a method for determining the correct number of clusters in a univariate time series and further, to develop a profitable pairs trading strategy based on Modern Portfolio Theory.

The rest of the thesis is structured as follows. Chapter 2 will review key definitions in multivariate time series analysis. Chapter 3 will review and summarise the k-means algorithm, the Maximum Mean Discrepancy, the p-Wasserstein distance and the WK-means algorithm as described in [1]. Chapter 4 will introduce the 1-dimensional WK-means algorithm and our extension to the d-dimensional space. Chapter 5 will showcase our experiments using synthetic data and the ability of our algorithm to detect changes in market regimes characterised by changes in the mean, variance and correlation of the underlying probability distribution used to generate the synthetic data. Chapter 6 will show the effectiveness of the algorithm when using real market data while chapter 7 will discuss how we may build profitable trading strategies from our findings. Finally, chapter 8 will conclude our work and discuss possible future research.

Chapter 2

Multivariate data and distributions

In this chapter we briefly review some key definitions related to multivariate time series and distributions.

Definition 2.0.1 (Time series, [36] (Section 1.2, page 2))**.** Suppose X is a random variable. If we observe X sequentially in time over or at a fixed interval, known as the sampling interval, the resulting data form a time series. We refer to these observations as realisations of X and a single observation as x. The time series is thus the collection of sequential observations $\{x_1, \ldots, x_n\}$ where n is the total number of observations.

When discussing univariate time series, we refer to a random variable X. If we were to take observations of the return of Apple stock at fixed intervals, say every hour, then these observations would form a univariate time series and the return of Apple is thus our random variable. We would then say that the return of Apple is distributed over some range of (possibly infinite) points and we might refer to the expectation and variance of this distribution.

Definition 2.0.2 (PDF and CDF of a random variable, [37] (Section 3.1, page 24))**.** Suppose X is a random variable. We say that X has probability density function (PDF), denoted $f_X(\cdot)$ over some interval (a, b) for $a, b \in \mathbb{R}$ if $f_X(\cdot)$ is a non-negative, Lebesgue measurable function given by

$$\mathbb{P}(a \leq x \leq b) = \int_a^b f_X(x)dx.$$

We say that the cumulative distribution function (CDF) is then given by

$$F_X(x) := \mathbb{P}(X \leq x) = \int_a^x f_X(t)dt.$$

Definition 2.0.3 (Expectation and variance of a random variable, [37] (Section 3.1, page 24))**.** Suppose X is a random variable on some interval $(a, b) \subset \mathbb{R}$. The expectation, $\mathbb{E}(X)$ or $\mu(X)$, over (a, b) is given by

$$\mathbb{E}(X) := \int_a^b x f_X(x)dx.$$

We define the variance, $\mathbb{V}(X)$ or $\sigma(X)^2$, to be

$$\mathbb{V}(X) := \int_a^b (x - \mu(X))^2 f_X(x)dx,$$

and the standard deviation $\sigma(X) := \sqrt{\mathbb{V}(X)}$.

Multivariate time series are an extension of the univariate case. Observations are made on a set of random variables $X_1, X_2, \ldots, X_d$ where d is the number of random variables. As in the

univariate case we might make n such observations, and we may then visualise these observations in matrix form as in 2.1.

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{d1} & x_{d2} & \cdots & x_{dn} \end{pmatrix}$$

Figure 2.1: Matrix form of a $d \times n$ multivariate time series.

In 2.1 we have d rows and n columns. Each row i might represent the observations of asset i, and each column j the observations of all d assets at time j. Analogously to the univariate case, we can define the PDF and CDF of a multivariate time series.

Definition 2.0.4 (Multivariate PDF and CDF, [37] (Section 3.3, page 28))**.** Suppose $X_1, \ldots, X_d$ are random variables. We say that $X_1, \ldots, X_d$ have a joint probability density function (PDF), denoted $f_{X_1,\ldots,X_d}(\cdot)$ over some interval $I \subset \mathbb{R}$ if $f_{X_1,\ldots,X_d}(\cdot)$ is a non-negative, Lebesgue measurable function given by

$$\mathbb{P}((x_1, \ldots, x_d) \in I) = \int_I f_{X_1,\ldots,X_d}(x_1, \ldots, x_d) dx_1 \cdots dx_d.$$

We say that the cumulative distribution function (CDF) is then given by

$$F_{X_1,\ldots,X_d}(x) := \mathbb{P}(X_1 \leq x_1, \ldots, X_d \leq x_d).$$

We refer to both the multivariate PDF and CDF interchangeably as the joint distribution of the random variables. A univariate distribution is typically characterised by its moments, the first two of which are the mean and variance as defined above. The joint distribution for multivariate data may be characterised by a range of possible characteristics including its marginal distributions, copula structure and correlation structure amongst others. In certain experiments we will focus on the correlation structure of our random variables.

Definition 2.0.5 (Covariance and Pearson correlation of a collection of random variables, [37] (Section 3.3, page 28))**.** Suppose $X_1, X_2, \ldots, X_d$ are random variables. The covariance of X_i and X_j for $i, j \in \{1, \ldots, d\}$ is given by

$$\mathrm{Cov}(X_i, X_j) := \mathbb{E}[(X_i - \mathbb{E}(X_i))(X_j - \mathbb{E}(X_j))].$$

Furthermore, we define the Pearson correlation of X_i and X_j as

$$\mathrm{Corr}(X_i, X_j) := \frac{\mathrm{Cov}(X_i, X_j)}{\sqrt{\mathbb{V}(X_i)}\sqrt{\mathbb{V}(X_j)}}.$$

In this paper we aim to study characteristics of the underlying distributions that govern the regimes of assets' returns. In previous work on the univariate case [1], the authors studied the mean and variance of the uni-dimensional distributions in order to categorise bull and bear market regimes. A bull market regime was characterised most notably by its lower variance and higher mean, and a bear market regime most notably by its higher variance and lower mean. In our work, we investigate whether we can capture regimes generated by the uni-dimensional distributions of the marginal distributions and those generated by changes in the joint distribution, such as changes in the correlation structure. We refer to changes in the mean and variance of the underlying marginal distributions as changes in *mean-variance regime* while changes in the correlation between the assets are referred to as changes in *correlation regime*. Collectively, we refer to changes in the underlying joint distribution as regime change. For clarity, we refer to the overall clustering problem as the *market regime clustering problem* (MRCP). In order to study these regimes, we must first define our method of choice: an adaption of the well-known k-means algorithm.

Chapter 3

Distance metrics and clustering algorithms

In this chapter we introduce the k-means clustering algorithm, the Maximum Mean Discrepancy, the p-Wasserstein distance metric and the WK-means algorithm. This section is partly an adaption of the original development of the WK-means algorithm in [1]. We begin with an introduction to various distance metrics and clustering algorithms.

3.1 Background

In this paper, we aim not to cluster the raw data from the multivariate time series of asset returns but to instead cluster a set of empirical measures to which this raw data is associated. Each cluster should then represent a different market regime to be found in our returns data. This clustering method requires the use of a distance metric that can be used to cluster probability distributions. A variety of such distance metrics are available for study.

The Kullback-Liebler (K-L) divergence and its symmetrical counterpart the Jensen-Shannon (J-S) divergence are popular divergences which have been employed with clustering problems in the literature before [38], [39]. Informally, given two probability distributions $\mathbb{P}$ and $\mathbb{Q}$, the K-L divergence is a measure of the similarity between the two probability distributions, measured as the expectation of the logarithmic difference between the two. Often $\mathbb{P}$ will represent a set of data while $\mathbb{Q}$ is a reference probability distribution we wish to compare the data to. It has been shown that the K-L divergence converges in distribution and total variation. One drawback of using either divergence is that they would require an estimation of the density functions of our empirical measures [40].

The Kolmogorov-Smirnov (K-S) test statistic uses the empirical distribution function in order to compare the distributions of two random variables. Informally, it finds the largest absolute distance between the two distributions across the supplied sample data. By the Glivenko-Cantelli theorem, if the empirical distribution functions are in fact the same, then the largest absolute distance between the two should decrease to zero as the number of samples increases. The K-S test statistic has been used with the k-means clustering algorithm in a financial setting [41] but it is known that it often lacks the sensitivity required to distinguish between elements of the clustering set [42].

The Maximum Mean Discrepancy (MMD) is another possible candidate that could be adopted when studying the similarity between empirical distributions. Informally, for i.i.d. sample data $(x_1, \ldots, x_N)$ and $(y_1, \ldots, y_N)$ from two discrete distributions $\mathbb{P}$ and $\mathbb{Q}$, the MMD is found as the supremum across a suitable set of functions $f \in \mathcal{F}$ of the difference in expectations of $f(x_i)$ and $f(y_i)$ for $i = 1, \ldots, N$. Similarly to the K-S test statistic, the MMD can be used as a two sample test and it has been used in the context of change point detection before [43], [44]. Recently, it has been combined with signatures in order to study the problem of regime detection and classification,

and outlier detection in a multidimensional setting [14].

Of course the distance metric chosen is only one aspect of our approach. Equally as important is the choice of clustering algorithm used. Many clustering algorithms have been developed to tackle problems in unsupervised learning. The fuzzy c-means algorithm is a close relative of the classic k-means algorithm. First proposed by J.C. Dunn in 1973 [45], the fuzzy c-means algorithm does not assign a data point to any one particular cluster but instead to all clusters with varying weights. Therefore, the fuzzy c-means algorithm may provide a more nuanced interpretation of clusters and may allow for a richer understanding of the membership properties of each data point.

The k-medoids algorithm is also a close relative of the k-means algorithm. Both algorithms partition a set of data into k clusters and both attempt to minimise some distance metric between the points in a given cluster. The two methods differ in certain key aspects however. The k-medoids algorithm takes its centroids from the data itself while the k-means algorithm often creates a new point as its centroid, one that is the mean of all points in a cluster. The k-medoids algorithm works with sums of pairwise dissimilarities while the k-means algorithm typically uses the sum of squared Euclidean distances. These differences often make the k-medoids algorithm more robust to noise and outliers when compared to the k-means algorithm [46].

Hierarchical clustering is a very different method of clustering. It typically comes in two flavours: agglomerative and divisive. The most common form is agglomerative. In agglomerative hierarchical clustering, each data point is first considered to be its own cluster and clusters are then combined at each step using a linkage criterion. Many linkage criterion have been suggested, each with their own uses, advantages and disadvantages. For example, single-linkage clustering forms new clusters by combining the closest pair of elements which do not yet belong to the same cluster. In contrast, Ward's linkage criterion suggests clustering based on the optimisation of some chosen function. Often, this function is the squared Euclidean distance and clusters are formed such that they minimise the variance between the clustered points. The hierarchical clustering algorithm does not require us to input the number of clusters as a parameter. Instead, given N data points, it will return k clusterings for k from 1 up to N. This can be advantageous when compared to the k-means clustering algorithm as often we may not know how many clusters a set of data should contain. As per proposition A.1.1, a good solution to the k-means algorithm requires that each cluster within our stream of data X is of roughly equal size. Versions of the hierarchical clustering algorithm do not require such a guarantee.

There are many more distance metrics and clustering algorithms available in addition to those that have been discussed, and a full review of the field could stretch into the hundreds of pages. In this paper, we build on the work of [1] and extend it to a multidimensional setting. Our initial choice of distance metric is therefore the p-Wasserstein distance and our choice of clustering algorithm is the k-means clustering algorithm. Calculation of the p-Wasserstein distance does not require us to find an approximation to the probability distribution function as in the case of the K-L and J-S divergences while we favour the k-means algorithm due to its observed speed [47] and relative simplicity. Drawing conclusions from the clusters yielded by k-means can also be easier than when using fuzzy c-means while hierarchical clustering can be slow since the standard agglomerative algorithm has a time complexity of $\mathcal{O}(N^3)$ and memory complexity of $\Omega(N^2)$. In addition to the p-Wasserstein distance, we will also test the Maximum Mean Discrepancy as our distance metric and compare the results of using each metric across a range of experiments.

Having briefly reviewed the vast world of distance metrics and clustering algorithms, we now discuss in more detail the k-means clustering algorithm, the Maximum Mean Discrepancy, the p-Wasserstein distance metric and the WK-means algorithm.

3.2 The k-means clustering algorithm

Definition 3.2.1 (Unsupervised learning, [48])**.** Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention.

The k-means clustering algorithm, or Lloyd's algorithm as it is also known, is a well-known and celebrated unsupervised learning algorithm that can be traced back to the work of J. MacQueen in [5]. As in many unsupervised learning algorithms, its use resides in identifying unseen patterns and clusters in the underlying data. The goal of the k-means clustering algorithm is to partition a collection of data into k distinct clusters.

Definition 3.2.2 (Set of data streams, [49] (Definition 2.1, page 3))**.** Take $\mathcal{X}$ to be a non-empty set. The set of streams of data over $\mathcal{X}$ is given by

$$\mathcal{S}(\mathcal{X}) := \{x = (x_1, \dots, x_n) : x_i \in \mathcal{X}, n \in \mathbb{N}\}.$$

Suppose we have a stream of data $X = (x_1, \dots, x_n) \in \mathcal{S}(V)$, over a normed vector space $(V, \|\cdot\|_V)$. In this paper we take $V = \mathbb{R}^d$ for $d \in \mathbb{Z}_+$. To each x_i, $i \in \{1, \dots, n\}$, we associate a tuple $x_i = (x_1^i, \dots, x_d^i)$. For example, if we take $d = 1$ we might consider x to be the price path of one particular asset with $V = \mathbb{R}$. If we take $d = 3$, we would then have three price paths such that at each increment i we have $x_i = (x_1^i, x_2^i, x_3^i)$ with x_1^i the price of asset 1 at index i, x_2^i the price of asset 2 at index i and x_3^i the price of asset 3 at index i. In this case, $V = \mathbb{R}^3$. We might also think of x as being a subset of the columns of our $M \times N$ matrix in figure 2.1.

In the case of $d > 1$, it is common practice to standardise the values of each x_i coordinate-wise, by finding the mean average and standard deviation of the values and subsequently defining a new stream of values $\hat{X} = (\hat{x}_1, \dots, \hat{x}_N)$, $\hat{x}_i \in \mathbb{R}^d$ such that

$$\mathbb{E}(\{\hat{x}_j^i\}_{1 \le i \le N}) = 0, \quad \mathbb{V}(\{\hat{x}_j^i\}_{1 \le i \le N}) = 1,$$

for $j = 1, \dots, d$. The elements of $\hat{X}$ are then assigned to k distinct clusters using an iterative algorithm, where k is a parameter we supply to the algorithm before starting.

Definition 3.2.3 (Set of clusterings over X, [1] (Definition A.1, page 29))**.** We write

$$\mathcal{C}(X) := \left\{ \{\mathcal{C}_i\}_{0 \le i \le n} : \mathcal{C}_i \cap \mathcal{C}_j = \emptyset, \cup_{i=1}^n \mathcal{C}_i = X, n \in \mathbb{N} \right\}$$

to be the set of all possible (disjoint) clusterings over X.

The goal of the algorithm is to return an element $\mathcal{C}^* \in \mathcal{C}(\hat{X})$ which is locally optimal with respect to some distance metric $D : \mathbb{R}^d \times \mathbb{R}^d \to [0, \infty)$ on $\mathbb{R}^d$. Often the distance metric used for clustering is the well-known Euclidean distance $D(x, y) = \|x - y\|$ where $x, y \in \mathbb{R}^d$. We note however that it is possible to use any distance metric on the relevant normed vector space. The general steps of the k-means clustering algorithm are as follows.

Step 1: Choose k initial centroids

Traditionally in the naive k-means algorithm we begin by choosing, uniformly at random, k points of $\hat{X}$, defined as $\bar{x} := \{\bar{x}_j\}_{1 \le j \le k}$, which we refer to as centroids. These points can be thought of as central points in each cluster, and it is around these centres that we cluster our remaining values. An alternative method is the k-means++ clustering algorithm [50].

In the k-means++ clustering algorithm, we choose one centroid randomly from the points in $\hat{X}$. For each remaining data point $\hat{x}_i$ in $\hat{X}$, we calculate the distance between it and the centroid that has already been chosen, denoted $D(\hat{x}_i)^2$. We then choose our next centroid using a weighted probability distribution. That is to say, the probability of choosing any given point $\hat{x}_i$ as the next centroid is

$$\frac{D(\hat{x}_i)^2}{\sum_{\hat{x}_j \in \hat{X}} D(\hat{x}_j)^2}.$$

We then repeat the preceding steps until we have k centroids. This method has been found to lead to faster convergence ($\mathcal{O}(log(k))$-competitive) with a higher degree of accuracy.

Step 2: Define the clusters

We then calculate the distance between each point $\hat{x}_i$ and each centroid in $\bar{x}$. Point $\hat{x}_i$ is then assigned to the j^{th} cluster for which its distance to the j^{th} centroid is smallest. More formally, we define the set of clusters such that

$$\mathcal{C}_j := \{\hat{x}_i \in \hat{X} : \text{argmin}_{l=1,\dots,k} D(\hat{x}_i, \bar{x}_l) = j\},$$

for $j = 1, \dots, k$.

Step 3: Define the new centroids

Finally, we define a new centroid for each cluster via a function $\alpha : 2^V \to V$ such that

$$\bar{x}_j := \alpha(C_j),$$

for $j = 1, \dots, k$. On d-dimensional space and for a given cluster $\mathcal{C}_j$ the function $\alpha(\cdot)$ is usually defined as:

$$\alpha(\mathcal{C}_j) := \left(\frac{1}{|\mathcal{C}_j|} \sum_{x \in \mathcal{C}_j} x_i\right)_{1 \leq i \leq d}.$$

Remark 3.2.4. In our experimental work we make use of the k-means++ algorithm. Irrespective of whether we use the naive k-means or k-means++ algorithm, we will refer to steps 1-3 as the k-means algorithm going forward.

Step 1 is a preliminary step completed only at the very start of the algorithm while, together, steps 2 and 3 form one iteration of the k-means clustering algorithm. Finally, we must define a stopping rule for the algorithm after which, the most recent clusters will be returned.

Definition 3.2.5 (k-means stopping rule, [1] (Definition A.4, page 30))**.** Suppose $(V, \|\cdot\|_V)$ is a normed vector space. For fixed $k \in \mathbb{N}$, consider a loss function $l : V^k \times V^k \to \mathbb{R}_+$ given by

$$l(x, y) := \sum_{i=1}^{k} \|x_i - y_i\|_V.$$

For a tolerance level $\epsilon > 0$, the stopping rule corresponding to the k-means algorithm is given by

$$l(\bar{x}^{n-1}, \bar{x}^n) < \epsilon,$$

where $n \in \mathbb{N}$ denotes the step of the algorithm, $V = \mathbb{R}^d$ for $d \in \mathbb{Z}_+$ and $\bar{x}^n$ is the collection of centroids at the n^{th} step of the algorithm.

In practice we may stop the algorithm should a certain number of iterations occur before full convergence is achieved. We recall that the goal of the algorithm is to find a set of clusters $\mathcal{C}^*$ which are locally optimal with respect to D, the distance metric. $\mathcal{C}^* \in \mathcal{C}(\hat{X})$ is thus simply a set of disjoint clusters which support $\hat{X}$ and it should be such that it minimises the total cluster variation.

Definition 3.2.6 (Within-cluster variation, [1] (Definition A.2, page 29))**.** Let $k \in \mathbb{N}$ and let $X \in \mathcal{S}(V)$ be a stream of data over a normed vector space V. Suppose $\mathcal{C} \subset \mathcal{C}(X)$ are disjoint clusters over X. Associate to each $\mathcal{C}_j$ its centroid $\bar{x}_j$ for $j = 1, \dots, k$. Then, for a given $\mathcal{C}_j$, the within-cluster variation is defined as

$$WC(\mathcal{C}_j) := \sum_{x \in \mathcal{C}_j} \|x - \bar{x}_j\|_V^2,$$

for $j = 1, \dots, k$.

Definition 3.2.7 (Total-cluster variation, [1] (Definition A.3, page 29))**.** With the notation of definition 3.2.6, define

$$TC(\mathcal{C}) := \sum_{j=1}^{k} WC(\mathcal{C}_j)$$

to be the total-cluster variation corresponding to a clustering $\mathcal{C} \in \mathcal{C}(X)$ on the normed vector space $(V, \|\cdot\|_V)$.

3.3 Maximum Mean Discrepancy

The Maximum Mean Discrepancy is an integrable probability metric, first proposed in [51]. It is introduced as a test statistic in the context of problem A.1.2. Formally it is defined as follows.

Definition 3.3.1 (Maximum Mean Discrepancy (MMD), [51] (Definition 2, page 726))**.** Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$ and let X and Y be random variables defined on a topological space $\mathcal{X}$. Let μ and ν be the Borel probability measures of X and Y respectively and suppose that x and y are independent samples drawn from X and Y. Then the Maximum Mean Discrepancy (MMD) between μ and ν is defined as

$$\mathrm{MMD}[\mathcal{F}, \mu, \nu] := \sup_{f \in \mathcal{F}} \left(\mathbb{E}_\mu[f(x)] - \mathbb{E}_\nu[f(y)] \right).$$

If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ where $x_i \sim \mu$ and $y_j \sim \nu$, then a biased empirical estimate of the MMD is given by

$$\mathrm{MMD}_b[\mathcal{F}, x, y] := \sup_{f \in \mathcal{F}} \left[\frac{1}{n} \sum_{i=1}^{n} f(x_i) - \frac{1}{m} \sum_{j=1}^{m} f(y_j) \right].$$

An important aspect to the MMD is the class of functions $\mathcal{F}$ over which it is defined. Although other candidates exist, kernel methods are often used to define $\mathcal{F}$. In [51] the unit ball in a reproducing kernel Hilbert space (RKHS), defined as a Hilbert space $\mathcal{H}$ (A.1.2) and an associated reproducing kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (A.1.3), is suggested for $\mathcal{F}$. When defined on a RKHS, if the choice of RKHS is universal (A.1.4), then the MMD can be shown to be a metric when defined on a compact space $\mathcal{X}$.

In the case of $\mathcal{X}$ being non-compact, then it has been shown that the MMD is still a metric if the kernel κ is a characteristic kernel [52] as defined in appendix A.1.5. The Gaussian kernel defined as

$$\kappa_G : \mathbb{R}^d \times \mathbb{R}^d \to [0, \infty], \quad \kappa_G(x, y) = \exp(-\|x - y\|^2_{\mathbb{R}^d} / 2\sigma^2),$$

for (x, y) in $\mathbb{R}^d \times \mathbb{R}^d$, is a characteristic kernel on the set of Borel measures on $\mathcal{X}$ [52]. We will use the MMD with $\mathcal{F} = (\mathcal{H}, \kappa_G)$ as one of our two distance metrics when clustering.

3.4 p-Wasserstein distance

The p-Wasserstein distance was first defined by Leonid Kantorovich in the context of optimisation methods [53]. It is oft-associated with the optimal transport problem and, where $p = 1$, the Earth mover's distance (EMD) [54]. Suppose we wish to move a pile of earth with shape μ into a hole of shape ν. This only makes sense if μ and ν have some equivalent mass. Suppose also that there is a cost associated with moving the pile from point x to point y called $D(x, y)^1 \geq 0$. We call a function $\mathbb{P}(x, y)$ a transport plan to move μ into ν if it describes the amount of mass to move from x to y. The 1-Wasserstein distance is then the minimal cost needed assuming the optimal transport plan is used.

Definition 3.4.1 (p-Wasserstein distance, [55] (Equation 7.1.1, page 151))**.** Suppose (X, D) is a separable Radon space and that $\mathcal{P}_p(X)$ is the set of probability measures on X with finite p^{th} moment. The p^{th} Wasserstein distance between measures μ, $\nu \in \mathcal{P}_p(X)$ is defined by

$$\mathcal{W}_p(\mu, \nu) := \left(\min_{\mathbb{P} \in \Pi(\mu,\nu)} \left\{ \int_{X \times X} D(x, y)^p \mathbb{P}(dx, dy) \right\} \right)^{\frac{1}{p}},$$

where

$$\Pi(\mu, \nu) := \{\mathbb{P} \in P(X \times X) : P(A \times X) = \mu(A), \mathbb{P}(X \times B) = \nu(B)\}$$

is the set of transport plans between μ and ν.

Proposition 3.4.2 (The p-Wasserstein distance is a metric, [56] (Chapter 5.1, proposition 5.4))**.** *The distance $\mathcal{W}_p : \mathcal{P}_p(X) \times \mathcal{P}_p(X) \to [0, \infty)$ is a metric on $\mathcal{P}_p(X)$.*

Proof. See appendix A.1.6 □

Proposition 3.4.2 states that the p-Wasserstein distance is indeed a metric, and we therefore can in fact use the p-Wasserstein distance as our distance metric in the k-means clustering algorithm. Furthermore, the concept of an average or central measure amongst a set of measures has a natural form when using the p-Wasserstein distance. This is called the Wasserstein barycentre.

Definition 3.4.3 (Wasserstein barycentre, [1] (Definition 2.3, page 9))**.** Suppose (X, D) is a separable Radon space and let $\mathcal{K} = \{\mu_i\}_{i \geq 1} \subset \mathcal{P}_p(X)$ be a family of Radon measures. The p-Wasserstein barycentre between measures $\bar{\mu}$ of $\mathcal{K}$ is defined to be

$$\bar{\mu} := \text{argmin}_{\nu \in \mathcal{P}_p(X)} \sum_{\mu_i \in \mathcal{K}} \mathcal{W}_p(\mu_i, \nu).$$

Remark 3.4.4. The p-Wasserstein distance defined in definition 3.4.1, is in fact a special case of an integral probability metric via its dual formulation and it shares a relationship with the MMD as shown in appendix A.1.7.

3.5 The WK-means algorithm

In this section we formalise our ideas thus far and explain how we can combine the k-means algorithm and p-Wasserstein distance in order to yield the WK-means algorithm as in [1]. We note that the goal of the market regime clustering problem (MRCP) can be stated as the task of classifying segments of return series $(r_i)_{i \geq 0}$ where

$$r_i = (r_i^1, \ldots, r_i^n),$$

for $n \in \mathbb{N}$ and $r_i^j \in \mathbb{R}^d$ for $j \in \{1, \ldots, n\}$, where $d \in \mathbb{Z}_+$. We note as well that for any vector $r_i \in \mathbb{R}^n$, we may form an empirical measure $\mu_{r_i} = \frac{1}{n} \sum_{j=1}^n \delta_{r_i^j}$, using the dirac delta function δ. Therefore, we may rewrite the goal of the MRCP as being equivalent to assigning labels to empirical probability measures $\mu \in \mathcal{P}_p(\mathbb{R}^d)$, where $\mathcal{P}_p(\mathbb{R}^d)$ is the set of probability measures on $\mathbb{R}^d$ with finite p^{th} moment. In practice we wish to take potentially overlapping segments of returns data for d assets and gather the empirical measures associated to these segments together into distinct clusters, each defined by an underlying distribution, which we then refer to as market regimes.

In [1], it was taken that in definition 3.2.2, $\mathcal{X} = \mathbb{R}$. In this paper, we will take $\mathcal{X} = \mathbb{R}^d$ for some $d \in \mathbb{Z}_+$. We fix $N \in \mathbb{N}$ and we will work with elements of the form $S = (s_0, \ldots, s_N) \in \mathcal{S}(\mathbb{R}^d)$, which are price paths of d financial assets. For $S \in \mathcal{S}(\mathbb{R}^d)$, we define the vector of log-returns r^S associated to S as

$$r_i^S = \log(s_{i+1}) - \log(s_i),$$

for $0 \leq i \leq N - 1$ and thus $r^S \in S(\mathbb{R}^d)$. As a firm example of these definitions, suppose we have two assets A and B such that $d = 2$, and such that the prices of A and B at times 0 and 1 are 20 and 22, and 30 and 34 respectively. Then we would have

$$S = (s_0, s_1),$$
$$s_0 = [20, 30]^T, s_1 = [22, 34]^T,$$
$$r_0^S = [\log(22) - \log(20), \log(34) - \log(30)]^T = [0.04, 0.05]^T.$$

In order to divide our original stream of returns data into segments we use definition 3.5.1.

Definition 3.5.1 (Stream lift, [49] (Section 3.3, page 5))**.** Let $\mathcal{S}(\mathbb{R}^d)$ be a space of streams over the $\mathbb{R}^d$ and let $m \geq 1$. We call a function

$$\ell = (\ell^1, \ldots, \ell^m) : \mathcal{S}(\mathbb{R}^d) \to \mathcal{S}(\mathcal{S}(\mathbb{R}^d))$$

a lift from the space of streams to the space of streams of segments over $\mathbb{R}^d$.

Using definition 3.5.1 we define a lift l that will take our original returns data of length N, and divide it into M segments of length h_1 using an overlap parameter of length h_2, where h_1, $h_2 \in \mathbb{N}$, $M := \lfloor \frac{N}{h_1 - h_2} \rfloor$ and $\ell := \ell_{h_1,h_2}$ such that

$$\ell^i(r^S) = (r^S_{(h_1-h_2)(i-1)}, \ldots, r^S_{h_1+(h_1-h_2)(i-1)}),$$

for $i = 1, \ldots, M$. We now have M segments of length h_1 for which h_2 returns overlap between each successive segment. As discussed, any segment of returns data $r_i \in \ell(r^S)$ with $r_i = (r_i^1, \ldots, r_i^{h_1})$ can be associated to an empirical measure $\mu_i = \frac{1}{n} \sum_{j=1}^n \delta_{r_i^j}$. Therefore we may define a family of measures

$$\mathcal{K} := \{(\mu_1, \ldots, \mu_M) : \mu_i \in \mathcal{P}_p(\mathbb{R}^d), i = 1, \ldots, M\}. \tag{3.5.1}$$

We now have a set of measures associated to our raw multivariate time series of returns data. In order to use the k-means algorithm with this family of measures, we need to make certain adaptions to the steps laid out in section 3.2. The overarching idea is that instead of using the traditional Euclidean distance as our distance metric, we use a different distance metric in order to measure the distance between measures μ and ν in $\mathcal{K}$.

In [1], the authors argue that the p-Wasserstein distance is the natural choice when clustering probability measures $\mathcal{P}_p(\mathbb{R})$ defined on the metric space (X, D), in particular for the univariate case of dimension $d = 1$. The distance function D appears directly in definition 3.4.1 and by proposition 3.5.2, the p-Wasserstein distance metrizes weak convergence.

Proposition 3.5.2 (Weak convergence of p-Wasserstein distance, [57] (Chapters 2 & 3, pages 1 - 155))**.** *A sequence $(\mu_i)_{\geq 1} \subset \mathcal{P}_p(X)$ converges weakly to $\mu \in \mathcal{P}_p(X)$ if and only if $\mathcal{W}_p(\mu_i, \mu) \to 0$ as $n \to \infty$.*

Proof. See Chapters 2 & 3, pages 1 - 155 in [57]. □

In the univariate case of dimension $d = 1$, the p-Wasserstein distance is quick to compute and the aforementioned barycentre (definition 3.4.3) acts as a natural aggregator whose computation is also efficient. Analogously to section 3.2, we go through the adaptions to the k-means clustering algorithm required in order to use the p-Wasserstein distance effectively.

Step 1: Choose k initial centroids

The initial step of choosing k initial centroids, $\bar{\mu} := \{\bar{\mu}_j\}_{1 \leq j \leq k}$ is similar to the standard k-means algorithm. If using the naive k-means clustering algorithm, then we choose uniformly at random k measures from our family of measures $\mathcal{K}$ to be our initial centroids. If using the k-means++ algorithm, then the distance metric used in the weighted probability distribution, D, is the p-Wasserstein distance, $\mathcal{W}_p$, while all other steps remain the same.

Step 2: Define the clusters

We then calculate the distance between each point μ_i and each centroid in $\bar{\mu}$. Point μ_i is then assigned to the j^{th} cluster for which its distance to the j^{th} centroid is smallest. More formally, we define the set of clusters such that for $j = 1, \ldots, k$

$$\mathcal{C}_j := \{\mu_i \in \mathcal{K} : \text{argmin}_{l=1,\ldots,k} \mathcal{W}_p(\mu_i, \bar{\mu}_l) = j\}.$$

Step 3: Define the new centroids

Finally, we define a new centroid for each cluster using the p-Wasserstein barycentre as defined in definition 3.4.3. This is prudent as the barycentre of a family of measures $\{\mu_i\}_{i\geq 1}$ is the measure $\bar{\mu}$ which minimises the within-cluster variation as defined in definition 3.2.6. By minimising the within cluster variation, the total-cluster variation is thus minimised, as defined in definition 3.2.7.

The final adaption is to the stopping rule and its loss function l. The adaption is quite natural wherein, instead of using the squared Euclidean distance to measure the distance between the old and new cluster centres, we simply replace this with the p-Wasserstein distance. We then stop the algorithm once the difference in the old and new centroid values after each iteration falls below a preset parameter ϵ. Thus we have the function $l : \mathcal{P}_p(\mathbb{R}^d)^k \times \mathcal{P}_p(\mathbb{R}^d)^k \to [0, \infty)$ defined as

$$l(\bar{\mu}^{n-1}, \bar{\mu}^n) := \sum_{i=1}^{k} \mathcal{W}_p(\bar{\mu}_i^{n-1}, \bar{\mu}_i^n). \tag{3.5.2}$$

Therefore, for a given value of $\epsilon > 0$, we terminate the algorithm at step n if $l(\bar{\mu}^{n-1}, \bar{\mu}^n) < \epsilon$.

Definition 3.5.3 (WK-means algorithm, [1] (Definition 2.7, page 11))**.** Let $\mathcal{K} \subset \mathcal{P}_p(\mathbb{R}^d)$ be a family of measures with finite p^{th} moment. We refer to the k-means clustering algorithm on $(\mathcal{P}_p(\mathbb{R}^d), \mathcal{W}_p)$, with aggregation method given by the p-Wasserstein barycentre and loss function given by equation (3.5.2) as the Wasserstein k-means algorithm or WK-means algorithm.

Remark 3.5.4. Although we state the adaptions to the k-means clustering algorithm in terms of the p-Wasserstein distance, we will in fact use the same framework with the MMD.

Chapter 4

The WK-means and MMDK-means algorithms

In this chapter, we show how the WK-means algorithm is implemented in practice for the univariate ($d = 1$) case, and the generic d-dimensional case ($d > 1$). We separate these into distinct parts as it is well-known that the p-Wasserstein distance in the univariate case is computationally easier than in the generic d-dimensional case. We also show how we may use the Maximum Mean Discrepancy in order to form a similar algorithm which we dub the d-dimensional MMDK-means algorithm.

4.1 The uni-dimensional WK-means algorithm

Our discussion of the univariate case is primarily an adaption of the work in [1]. For univariate data we work on the space $(\mathcal{P}_p(\mathbb{R}), \mathcal{W}_p)$. In this case, given absolutely continuous measures $\mu, \nu \in \mathcal{P}_p(\mathbb{R})$, there exists a closed-form solution to the p-Wasserstein distance.

Proposition 4.1.1 ([58] (Equation (3), page 2))**.** *Suppose $\mu, \nu \in \mathcal{P}_p(\mathbb{R})$ and that μ, ν are absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}$. Then the p-Wasserstein distance $\mathcal{W}_p(\mu, \nu)$ is given by*

$$\mathcal{W}_p(\mu, \nu) = \left(\int_0^1 D(F_\mu^{-1}(z), F_\nu^{-1}(z))^p dz \right)^{\frac{1}{p}},$$

where the quantile function $F_\mu^{-1} : [0, 1) \to \mathbb{R}$ is defined as

$$F_\mu^{-1}(z) := \inf\{x : F_\mu(x) > z\}. \tag{4.1.1}$$

and $D(x, y) = |x - y|$.

Proof. See appendix A.2.1. □

The closed form solution given in proposition 4.1.1 makes computation of the p-Wasserstein distance in the univariate case efficient. To illustrate this, suppose we have two empirical measures $\mu, \nu \in \mathcal{P}_p(\mathbb{R})$. We recall the formal definition of an empirical measure in the uni-dimensional case.

Definition 4.1.2 (Empirical measure, [59] (Section 3.9.5, page 65))**.** Let $x \in \mathcal{S}(\mathbb{R})$ such that $x = (x_1, \ldots, x_N)$ for $N \in \mathbb{N}$. Furthermore, let

$$\mathcal{Q}_i : \mathcal{S}(\mathbb{R}) \to \mathbb{R}$$

be the function which extracts the i^{th} order statistic of x, for $i = 1, \ldots, N$. Then the cumulative distribution function of the empirical measure $\mu^x \in \mathcal{P}_p(\mathbb{R})$ associated to x is defined as

$$\mu^x((-\infty, x]) := \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\{\mathcal{Q}_i(x) \leq x\}}(x),$$

where $\mathbb{1} : \mathbb{R} \to [0, 1]$ is the indicator function. We refer to $(\mathcal{Q}_i(x))_{1 \leq i \leq N} = (\mathcal{Q}_i)_{1 \leq i \leq N}$ as the increasing sequence of atoms of μ^x.

We assume that our measures μ and ν each have an equal numbers of atoms $h_1 \in \mathbb{N}$ such that

$$\mu((-\infty, x]) = \frac{1}{h_1} \sum_{i=1}^{h_1} \mathbb{1}_{\{\alpha_i \leq x\}}(x), \quad \nu((-\infty, x]) = \frac{1}{h_1} \sum_{i=1}^{h_1} \mathbb{1}_{\{\beta_i \leq x\}}(x),$$

where $(\alpha_i)_{1 \leq i \leq h_1}$ and $(\beta_i)_{1 \leq i \leq h_1}$ are the increasing sequences of atoms for μ and ν respectively.

Definition 4.1.3 (Locally finite Borel measure, [60] (Page 55))**.** A locally finite Borel measure is a measure, μ, defined on the Borel σ-algebra such that every compact set has finite measure, that is

$$\mu(K) < \infty,$$

for every compact set K.

Definition 4.1.4 (Radon measure, [61] (Page 212))**.** Let X be a locally compact Hausdorff space. A Radon measure on X is a Borel measure that is finite on all compact sets, outer regular on all Borel sets, and inner regular on all open sets.

Theorem 4.1.5 ([62] (Theorem 4.3))**.** *There is a one-to-one correspondence between Radon measures μ on $[0, \infty)$ and right continuous functions $A_t : \mathbb{R} \to [0, 1]$ of finite variation given by*

$$A_t = \mu((-\infty, t)).$$

Every empirical measure on $\mathbb{R}$ is a locally finite Borel measure and is thus, in fact, Radon and hence by theorem 4.1.5 one can associate to μ (and similarly ν) a right-continuous function of finite variation $A_t : \mathbb{R} \to [0, 1]$ given by $A_t = \mu((-\infty, t))$. The inverse of this function exists and is given by the quantile function (4.1.1). Therefore, in the case of μ, we have that

$$F_\mu^{-1}(z) = \alpha_i, \quad z \in \left[\frac{i-1}{h_1}, \frac{i}{h_1}\right), \tag{4.1.2}$$

for all $i = 1, \ldots, h_1$ and analogously for ν. Using the fact that $F_\mu^{-1}(z) = 0$ for all $z < \alpha_1$, and proposition 4.1.1 combined with result (4.1.2), we have that

$$\begin{aligned} \mathcal{W}_p(\mu, \nu)^p = \int_0^1 D(F_\mu^{-1}(z), F_\nu^{-1}(z))^p dz &= \sum_{i=1}^{h_1} \int_{\frac{i-1}{h_1}}^{\frac{i}{h_1}} D(F_\mu^{-1}(z), F_\nu^{-1}(z))^p dz \\ &= \frac{1}{h_1} \sum_{i=1}^{h_1} D(\alpha_i, \beta_i)^p. \end{aligned} \tag{4.1.3}$$

We therefore have that, in the uni-dimensional case at least, calculating the p-Wasserstein distance is simply a matter of using a quick sort algorithm ($\mathcal{O}(log(h_1))$) to order each measure's atoms before a simple summation of distances is performed. Furthermore, the computation of the p-Wasserstien barycentre is efficient.

Proposition 4.1.6 (Uni-dimensional p-Wasserstein barycentre, [1] (Proposition 2.6, page 11))**.** *Suppose that $\{\mu_i\}_{1\leq i\leq M}$ are a family of empirical probability measures, where each measure has N atoms, $(\alpha_j^i)_{1\leq j\leq N}\subset\mathbb{R}^N$. Let*

$$a_j = Median(\alpha_j^1,\dots,\alpha_j^M), \quad b_j = Mean(\alpha_j^1,\dots,\alpha_j^M),$$

for $j=1,\dots,35$. Then the cumulative distribution function of the 1-Wasserstein barycentre $\bar{\mu}\in\mathcal{P}_1(\mathbb{R})$ over $\{\mu_i\}_{1\leq i\leq M}$ with respect to the 1-Wasserstein distance is given by

$$\bar{\mu}((-\infty,x]) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{\{a_i\leq x\}}(x),$$

and the p-Wasserstein barycentre $\bar{\mu}\in\mathcal{P}_p(\mathbb{R})$ over $\{\mu_i\}_{1\leq i\leq M}$ with respect to the p-Wasserstein distance, for $p>1$, is given by

$$\bar{\mu}((-\infty,x]) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{\{b_i\leq x\}}(x).$$

Moreover, $\bar{\mu}$ is not necessarily unique.

Proof. See appendix A.2.2. □

Therefore computation of the p-Wasserstein barycentre in the uni-dimensional case requires only calculation of a median or mean. Together, these results make the WK-means clustering algorithm efficient in the univariate case.

4.2 The d-dimensional WK-means algorithm

This section builds on the framework of the WK-means algorithm as presented in chapter 3. It uses different methods to that of the uni-dimensional implementation however. We work on the space $(\mathcal{P}_2(\mathbb{R}^d),\mathcal{W}_2)$, $d>1$ going forward.

4.2.1 Background

Computation of the p-Wasserstein distance and barycentre in $\mathbb{R}^d$, $d>1$, is known to be computationally demanding for h_1 or M large [63], with evaluation between multi-dimensional measures often numerically intractable [58]. In recent years, an alternative approach has been to use variations of the Wasserstein distance including the sliced Wasserstein distance, the max sliced Wasserstein distance and the generalized sliced Wasserstein distance. The general principle of the sliced Wasserstein distance is to obtain a family of one-dimensional representations of the original higher-dimensional probability distribution. This is done by projecting from the higher-dimensional probability distribution to the unit sphere. One then subsequently calculates the distance between two input distributions as a functional on the Wasserstein distance of their one-dimensional representations. This is typically achieved by using linear projections such as the Radon transform. The max and generalized versions are then variations of this wherein we use the maximum representation [64] (definition 2.1, page 3) and the generalized Radon transform respectively [58] (part 3.1, page 4). All three have been shown to be distance metrics [65] (proposition 5.1.2, page 120), [58] (part 3.2, page 5).

In using these variations of the Wasserstein distance, we can approximate the true d-dimensional Wasserstien distance using the uni-dimensional case which is inherently quicker. Furthermore, the sliced and max sliced p-Wasserstein distances have been shown to be equivalent to the p-Wasserstein distance for all $p\geq 1$ [64] (theorem 2.1, page 4) and the max sliced Wasserstein distance has been shown to be strongly equivalent to the p-Wasserstein distance for $p=1$ [64] (theorem 2.1, page 4) and $p=2$ [66] (proposition 2, page 4).

The Wasserstein barycentre in higher dimensions is often difficult to compute if not completely infeasible. Many algorithms have been proposed to help overcome this problem. These include a mixture of fixed support methods, where the atoms of the barycentre are fixed in advance and their optimal weightings need only be found, and free support methods, where the atoms and weightings are computed. A number of methods rely on some form of entropy regularisation [67], [68] while others use tools such as Bregman projections [69], stochastic gradient descent methods [63] and interior point methods [70].

Whether using the sliced versions of the Wasserstein distance or the barycentre approximation methods, in each case we are trading off some level of accuracy for speed. Rather than utilising these approximations, we instead return to the original problem itself.

4.2.2 Approach

In the next few paragraphs we lay out our approach for implementing steps 1-3 of the WK-means algorithm when working in d-dimensional space, which balances trade-offs in accuracy and speed. We recall that each empirical measure μ_i is formed from a segment of returns data $r_i \in \ell(r^S)$ with $r_i = (r_i^1, \ldots, r_i^{h_1})$ and hence μ_i is in fact a discrete density distribution in $\mathbb{R}^d$. Suppose we take two measures μ_i and μ_j from $\mathcal{K}$ (3.5.1) formed from segments $r_i \in \ell(r^S)$ with $r_i = (r_i^1, \ldots, r_i^{h_1})$ and $r_j \in \ell(r^S)$ with $r_j = (r_j^1, \ldots, r_j^{h_1})$ respectively, for $i, j \in 1, \ldots, M$.

We describe the elements of r_i and r_j, and thus the atoms of μ_i and μ_j respectively, as being point clouds X and Y in $\mathbb{R}^d$ of size h_1. As discussed in [63] (part 2.1, page 3) we may consider the 2-Wasserstein distance between X and Y as

$$\mathcal{W}_2(\mu_i, \mu_j)^2 = \mathcal{W}_2(\mathrm{X}, \mathrm{Y})^2 = \min_{\pi \in \Pi_{h_1}} \sum_{i \in I} \|\mathrm{X}_i - \mathrm{Y}_{\pi(i)}\|^2,$$

where Π_{h_1} is the set of all permutations of h_1 elements and $I = \{1, \ldots, h_1\}$. This distance is then found by computing the optimal assignment $i \to \pi^*(i)$ that minimises $\mathcal{W}_2(\mathrm{X}, \mathrm{Y})^2$. The problem can be viewed as a linear programming problem such that we aim to find

$$\mathcal{W}_2(\mathrm{X}, \mathrm{Y})^2 = \min_{B \in \mathcal{B}_{h_1}} \sum_{i,j \in I^2} B_{i,j} \|\mathrm{X}_i - \mathrm{Y}_j\|^2,$$

where $\mathcal{B}_{h_1}$ is the set of bistochastic matrices - that is, nonegative matrices with rows and columns summing to 1. This can be solved using the Hungarian algorithm [71] which is worst-case $\mathcal{O}(h_1^3)$ or, as shown in [72] (chapter 4), in $\mathcal{O}(h_1^{2.5} log(h_1))$ using dedicated solvers. The computational complexity of these algorithms typically makes this problem unattractive to solve for large h_1 or M. However, in python, use of the cython *cdist* function from the *scipy.spatial* package when computing the euclidean distance $\|\mathrm{X}_i - \mathrm{Y}_i\|^2$ and the cython *linear_sum_assignment* function from the *scipy.optimize* give an acceptable speed of computation for our experiments.

Computation of the 2-Wasserstein distance is required for all three steps of our iterative algorithm. In step one, we must compute the distance between each measure and its nearest centroid when using the k-means++ starting centroids. In step two, the distance between each measure and each centroid must be found in order to assign each measure to a cluster. Finally, in step three, we find the 2-Wasserstein barycentre of each cluster, a method which naturally requires computation of the 2-Wasserstein distance. In fact, the speed of computation is particularly important when it comes to solving for the 2-Wasserstein barycentre.

As discussed in section 4.2.1, computation of the Wasserstein barycentre can be difficult if not infeasible. In order to reduce the computational complexity, we restrict the possible atoms of the barycentre to those of the measures it is clustering. If the data is particularly sparse, this may lead to incorrect barycentres but assuming we work with a sufficient size of returns data, we reason that it should be reasonably accurate. Having restricted our search space, we then form an $M \times M$ distance matrix W, where the i^{th}, j^{th} entry is the 2-Wasserstein distance between the i^{th} and j^{th} measures.

$$\mathrm{W}_{ij} := \mathcal{W}_2(\mu_i, \mu_j) \to \begin{pmatrix} \mathrm{W}_{11} & \mathrm{W}_{12} & \cdots & \mathrm{W}_{1M} \\ \mathrm{W}_{21} & \mathrm{W}_{22} & \cdots & \mathrm{W}_{2M} \\ \vdots & \vdots & \cdots & \vdots \\ \mathrm{W}_{M1} & \mathrm{W}_{M2} & \cdots & \mathrm{W}_{MM} \end{pmatrix} \to \begin{pmatrix} \sum_j \mathrm{W}_{1j} \\ \sum_j \mathrm{W}_{2j} \\ \vdots \\ \sum_j \mathrm{W}_{Mj} \end{pmatrix} \to$$

$$q := \mathrm{argmin}_i \Big\{ \sum_j \mathrm{W}_{ij} \Big\},$$

$$\bar{\mu} := \mu_q$$

Figure 4.1: Barycentre computation flow diagram

Such a matrix is naturally symmetric which can be used to further cut down our computation time. We also know that $\mathcal{W}_{ij} = 0$ where $i = j$, and thus $\mathrm{W}_{ii} = 0$ for $i = 1, \ldots, M$. Using the *cdist* and *linear_sum_assignment* functions to find the 2-Wasserstein distance, $\mathcal{W}_2(\mu_i, \mu_j)$, as discussed, we form our distance matrix, which is $\mathcal{O}(\frac{M(M-1)}{2})$, and proceed to sum the rows. The row with the minimum sum is judged to be the entry of the measure which minimises the distance to all other measures in the cluster and is hence the 2-Wasserstein barycentre. Figure 4.1 shows a flow diagram of the described steps.

4.3 The d-dimensional MMDK-means algorithm

In the final part of this chapter we discuss how we may combine the Maximum Mean Discrepancy and k-means clustering algorithm to form the d-dimensional MMDK-means algorithm. For our purposes, we work on the space $(\mathcal{P}_2(\mathbb{R}^d), \mathrm{MMD})$, $d > 1$ going forward.

4.3.1 Approach

Our approach when using the MMD as the distance metric is analogous to that of the d-dimensional WK-means algorithm. An important difference between the two is that the MMD is often easier to compute in higher dimensions. When working with discrete distributions, we compute an unbiased estimate of the squared MMD as laid out in lemma 4.3.1.

Lemma 4.3.1 (Unbiased empirical estimate of the squared MMD, [51] (Lemma 6, page 728))**.** *Using the notation of 3.3.1, let x and x' be samples of X with distribution μ, and y and y' be samples of Y with distribution ν. The squared MMD is given by*

$$MMD^2[\mathcal{F}, \mu, \nu] = \mathbb{E}_{x,x'}[\kappa(x, x')] - 2\mathbb{E}[\kappa(x, y)] + \mathbb{E}_{y,y'}[\kappa(y, y')],$$

where x' and y' are independent copies of x and y respectively. An unbiased empirical estimate is given by

$$MMD_u^2[\mathcal{F}, X, Y] = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} \kappa(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \kappa(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \kappa(x_i, y_j).$$

Our choice of kernel function is the Gaussian kernel laid out in equation (3.3), a choice which requires a bandwidth parameter σ. The optimum choice of kernel size is an area of active research and it remains a heuristic. A number of Bandwidth estimation methods are discussed in detail in [73]. One suggestion proposed in [51] is to take σ to be the median distance between points in the aggregate sample. Overall the choice of bandwidth is data dependent. We found that averaging across a number of choices of bandwidth parameter on the scale of our constituent data worked well in our experiments, as opposed to instances of our bandwidth parameter being much larger than our data, which led to poor results.

Computation of a barycentre when working with the MMD is also required for our algorithm. Here we follow the same framework as the d-dimensional WK-means algorithm and define the barycentre to be the measure in a given cluster which minimises the distance to all other measures. Formally we define the barycentre to be

$$\bar{\mu}_j := \text{argmin}_{\nu \in \mathcal{C}_j} \sum_{\mu_i \in \mathcal{K}} \text{MMD}(\mu_i, \nu),$$

where $\mathcal{C}_j := \{\mu_i \in \mathcal{K} : \text{argmin}_{l=1,\dots,k} \text{MMD}(\mu_i, \bar{\mu}_l) = j\}$, a definition analogous to that of the p-Wasserstein case (definition 3.4.3).

Remark 4.3.2. Going forward, we will refer to our new algorithms as the 2-d WK-means algorithm and the 2-d MMDK-means algorithm respectively, and to the uni-dimensional WK-means algorithm as either the uni-d 1-WK-means algorithm or the uni-d 2-WK-means algorithm. We do this in order to highlight both the dimensionality of the data we run each algorithm on, and the choice of p used for the p-Wasserstein distance in each uni-dimensional algorithm.

Chapter 5

Synthetic data experiments

In this section, we test the 2-d WK-means and 2-d MMDK-means algorithms on synthetic data using two time series generated from either a classic geometric Brownian motion (GBM) or a Merton jump diffusion (MJD) process. We treat these time series as being synthetic price series and we convert this original price series into a stream of segments of returns as described in definition 3.5.1.

Testing unsupervised learning algorithms on real data can often be tricky given we do not know in advance the exact relationships that exist in the data. When clustering real 2-dimensional returns series, it is not possible to infer the exact underlying probabilistic structure from which they are generated and it is therefore difficult to define what exactly a 'correct' clustering is. This makes any tests conducted on real returns series hard to evaluate since it is impossible to know exactly when the underlying distribution has changed and thus when a regime change has occurred.

Therefore, in order to evaluate the accuracy of the clustering algorithm, we first tested it on synthetically generated data. This data was generated such that we knew *a priori* when shifts in the underlying probabilistic structure would occur in our returns data. This then allowed us to evaluate the true accuracy of the clusters formed.

The synthetic data generated from the GBM and MJD processes have a shared schema. Given $T \in \mathbb{N}$ and a time interval $[0, T]$, we define a mesh with increments that roughly represent one market hour. Taking $n := 252 \times 7$ to be the number of market hours in a market year, we set

$$\Delta := \left\{ \left[\frac{i-1}{n}, \frac{i}{n} \right] : i = 1, 2, \ldots, n \times T \right\}.$$

Therefore, T can be thought of as the number of market years. When generating our data using this mesh, the majority of points will fall into what we designate as a 'normal' regime with a particular underlying distribution.

At certain points this regime will change. Our goal is to study changes in the joint distribution of our assets, and so these changes will be changes of the mean and variance of the synthetic assets, of their correlation structure or a mixture of both. Each experiment will explicitly define its parameters and types of regime change. For a given experiment, we have J independent types of regime change and we define those periods of change as follows. We take $r^j \in \mathbb{N}$ to be the number of regime changes of type j for $j \in [1, J]$ that we wish to observe. Taking the price series of a synthetic asset to be $\{s_i\}_{1 \leq i \leq N}$, we specify their starting points and lengths by $(s_i^j, l_i^j) \in \mathbb{N} \times \mathbb{N}$ for $i = 1, \ldots, r^j$, such that

$$0 \leq s_0^j < s_{r^j}^j + l_{r^j}^j \leq n \times T,$$

and

$$s_i^j + l_i^j < s_{i+1}^j,$$

for $i = 1, \ldots, r^j - 1$. Each l_i^j may be a constant or a random variable but for our experiments, we kept this value fixed. We thus have a set of disjoint intervals representing our periods of regime

change of type j

$$R^j := \{[s_i^j, s_i^j + l_i^j] : i = 1, \dots, r^j\}.$$

We designate periods of non-regime change as $N := \Delta \setminus \cup_{j \in J} R^j$. Therefore periods of regime change will start at point s_i^j and end at point $s_i^j + l_i^j$ for each i in $[1, r^j]$ and j in $[1, J]$.

Using the mesh structure above, we generated two synthetic price series S_1 and S_2, and calculated the log-returns $r^S \in \mathcal{S}(\mathbb{R}^2)$ where $S := (S_1, S_2)$. The log-returns r^S have a joint distribution f_{S_1,S_2} and marginal distributions given by f_{S_1} and f_{S_2}. In line with (3.5.1), we created a lifted stream of data $l(r^S) \in \mathcal{S}(\mathcal{S}(\mathbb{R}^2))$, and subsequently a family of empirical measures $\mathcal{K}$, by segmenting our return series into M segments.

In order to create the lifted stream, we first chose a value for h_1 and h_2. In [1], when testing the original univariate WK-means algorithm, the authors choose $h_1 = 35$ and $h_2 = 28$ given $h_1 = 35$ corresponds to approximately one market week in market hours with h_2 being one market day short of that. The value of h_1 dictates the number of returns that form each empirical distribution while h_2 dictates the degree of overlap between segments. While one may reasonably want h_1 to be as large as possible in order to better approximate empirically the true underlying distribution of a segment, if h_1 is too large then we will likely miss certain regime changes. If the value is too small however then our classifications are likely to be dominated by noise. The value of h_2 can be used to increase the degree of overlap between the segments, which can be useful when our data is limited in size. In testing their choice of hyperparameters with real data, the authors conclude the values of $h_1 = 35$ and $h_2 = 28$ to be reasonable, and they note that for a suitably chosen h_1 and data of a sufficient size, the value of h_2 does not have a large effect on the centroids obtained. Moving forward, we will take $h_1 = 35$ and $h_2 = 28$ for the uni-d p-WK-means, 2-d WK-means and 2-d MMDK-means algorithms for our synthetic examples. We will further discuss the values of h_1 and h_2 for the 2-d WK-means and 2-d MMDK-means algorithms in the context of real data in chapter 6.

Remark 5.0.1. We note that given the choice of hyperparameters, and in particular due to the choice of overlap parameter $h_2 = 28$, any unique return may belong to between 1 and 5 segments.

We ran the uni-d p-WK-means and 2-d algorithms over the lifted stream of data $l(r^S)$, returning k centroids $\{\bar{\mu}_i\}_{i=1,\dots,k}$ and clusters $\{\mathcal{C}_j\}_{j=1,\dots,k}$. We display our results primarily using a combination of three different plots. The first is referred to as the mean-variance plot. Given each empirical joint distribution $\mu \in \mathcal{K}$ is formed from a multivariate time series and is a discrete distribution, it may be written in the form given in figure 5.1.

$$\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1h_1} \\ r_{21} & r_{22} & \cdots & r_{2h_1} \end{pmatrix}$$

Figure 5.1: Matrix form of a $2 \times h_1$ empirical distribution μ.

Each row is thus a marginal distribution of the joint distribution and each marginal distribution will have an associated mean and variance. Therefore each marginal distribution can be projected onto $\mathbb{R}^2$ via the mapping

$$M_p : \mathcal{P}_p(\mathbb{R}) \to \mathbb{R}^2,$$

$$\mu_{f_{S_i}} \to \left(\sqrt{\mathbb{V}(\mu_{f_{S_i}})}, \mathbb{E}(\mu_{f_{S_i}}) \right),$$

for $i = 1, 2$, which is simply a scatter plot of each measure in mean-variance space. Each point is then colored according to its cluster membership. Using this mean-variance plot allows us to visualise how well our algorithms cluster along each of the marginal distributions f_{S_1} and f_{S_2}.

The second plot we make use of is a plot of the log-returns of each asset which aids us in visualising the correlation between the points in each cluster. We identify each cluster, and its points therein, with a color. We recall that the empirical distributions formed from our returns series overlap. Given the returns series $r^S = (r_1^S, \dots, r_{N-1}^S)$, each return r_i^S, and thus asset price, may be associated with up to 5 different empirical measures and therefore, there are a number of potential centroid membership combinations, and colors, that it may have.

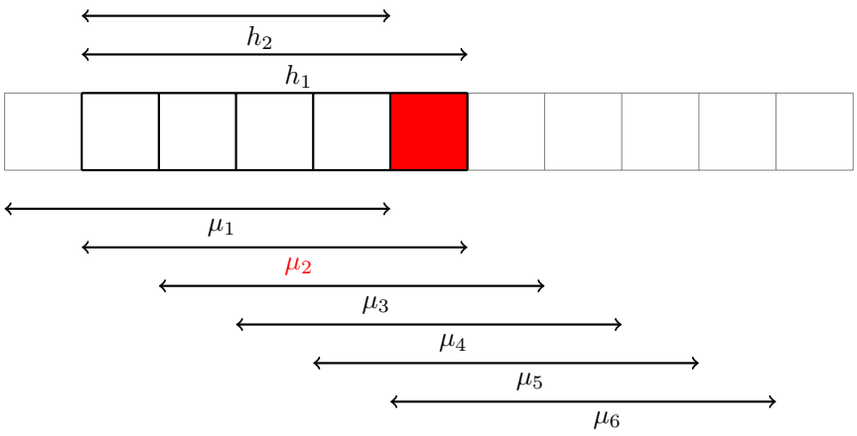


Figure 5.2: Example diagram of the segmentation of returns and the subsequent association of returns with a cluster color.

We aim to use this algorithm in the context of trading and hence, for the purposes of any color plots formed, we will associate each return r_i^S with the first centroid membership it has. This will also make our plots clearer, particularly when we have more than two colours in the same plot. This idea is demonstrated in 5.2 which shows an example of the segmented returns $l(r^S)$. Each block is of size $h_1 - h_2$, and each measure μ_i is of length h_1. Each measure μ_i overlaps with the preceding and subsequent measures. Therefore, given $h_1 = 35$ and $h_2 = 28$ the block highlighted in red is included in measures μ_2, μ_3, μ_4, μ_5, and μ_6. However, we give it the color of the cluster in which μ_2 is placed, which is highlighted with a thick border in the diagram. In this example, μ_2 was placed in the red cluster hence the returns in this block are colored red. The correlation plot is then formed by coloring each pair of returns $r_i^S = (r_i^{S_1}, r_i^{S_2})$ for $i \in \{1, \dots, N-1\}$, and plotting these returns pairs. This plot should show the correlation of the points in each cluster.

Finally, our third plot will be a colored time series plot. In a similar manner to the second plot, we partition the two price series S_1, S_2 such that each partition is colored according to its centroid membership. The centroid membership is again the first centroid to which a particular price can be associated. We will also apply a colored shade to the grid points indicating the periods of regime change which should allow us to visually ascertain how well the algorithm is picking up periods of regime change.

In order to numerically evaluate the accuracy of our clustering algorithm, we follow the approach of [1] and consider three scores of accuracy: total accuracy, accuracy during the normal regime (regime-off) and accuracy during periods of regime change (regime-on). We calculate these in the following way: for $i \in \{1, \dots, N-1\}$, associate to each log-return r_i^S the empirical measures $M_i := \{\mu_{j(i)}, \dots, \mu_{j(i)+v}\}$ that it is a member of where $j \in \mathbb{N}$ is the first such measure. We note that $v \in [0, 4]$ as seen in remark 5.0.1. We then calculate which cluster in $\{1, \dots, k\}$ each $\mu \in M_i$ is associated to, which gives us a set of labels $\bar{y}^i := \{\bar{k}_0, \dots, \bar{k}_v\}$. These values are then aggregated into a row vector

$$\bar{Y}^i := \left(\sum_{j=0}^{v} \mathbb{1}_{\{x=l\}}(\bar{k}_j) \right)_{l=1}^{k},$$

for $i = 1, \dots, N-1$. Going forward, we assume that $\bar{k} = 0$ corresponds to the standard regime and $\bar{k} = j$ corresponds to the j^{th} type of regime change for j in $[1, J]$. In our experiments we test regime change of the underlying distribution using a change of mean and variance, a change of correlation structure and a mixture of both changes. Therefore we have a number of different types of regime change and thus the accuracy during these periods will be dictated by the type of regime change which has occurred. Nonetheless, we can create a general definition for evaluation purposes.

Definition 5.0.2 (Accuracy scores, [1] (Definition 3.7, page 19))**.** With the notation previously given, for a given vector of log-returns $r^S \in \mathcal{S}(\mathbb{R}^d)$ and cluster assignments $\mathcal{C} = \{\mathcal{C}_j\}_{j=1}^k$, the regime-off accuracy score (ROFS) is given by

$$\text{ROFS}(r^S, \mathcal{C}) := \frac{\sum_{r_i^S \in N} \bar{Y}_0^i}{\sum_{r_i^S \in N} \sum_k \bar{Y}_k^i}.$$

Similarly, the regime-on accuracy score (RONS) for the j^{th} type of regime change is given by

$$\text{RONS}_j(r^S, \mathcal{C}) := \frac{\sum_{r_i^S \in R^j} \bar{Y}_j^i}{\sum_{r_i^S \in R^j} \sum_k \bar{Y}_k^i}.$$

Finally, total accuracy (TA) is given by

$$\text{TA}(r^S, \mathcal{C}) := \frac{\sum_{r_i^S \in N} \bar{Y}_0^i + \sum_{r_i^S \in R^1} \bar{Y}_1^i + \cdots + \sum_{r_i^S \in R^J} \bar{Y}_J^i}{\sum_{i=1}^N \sum_k \bar{Y}_k^i}.$$

We briefly outline the types of joint regime (JR) that are theoretically possible in our experimental framework. Given we are working with a combination of two assets, there are several types of joint regime possible. These are summarised in table 5.1. Each joint regime is a combination of the mean-variance regime and the correlation regime. The mean-variance regime of a particular joint regime is dictated by the mean-variance regime of each underlying asset. This is a combination of a lower variance/higher mean (bull) and higher variance/lower mean (bear) market for each asset. The correlation regime is also one of two. We refer to the correlation regime as being either the 'normal' correlation regime or the 'abnormal' correlation regime. We refer to the combination of JR = (Bull, Bull, Normal) as being the standard regime. The number of joint regimes in each of our experiments will dictate the number of clusters k we test for.

Asset 1	Asset 2	Correlation
Bull	Bull	Normal
Bull	Bear	Normal
Bear	Bull	Normal
Bear	Bear	Normal
Bull	Bull	Abnormal
Bull	Bear	Abnormal
Bear	Bull	Abnormal
Bear	Bear	Abnormal

Table 5.1: Possible combinations of joint regimes (JR).

An extensive number of experiments were conducted and in the following sections, we highlight a few key experiments and their findings.

Remark 5.0.3. In the following experiments, where clustering graphs have been provided these have been generated using the 2-d WK-means algorithm. Where statements are made regarding these graphs, similar statements may also be attributed to graphs generated using the 2-d MMDK-means algorithm where regime accuracy scores are similar.

5.1 Geometric Brownian motion

In this section we generate two synthetic price series modelled as geometric Brownian motions. Let $\mathcal{M}(\theta)$ be a family of models indexed by a parameter set $\theta \subset \mathbb{R}^5$. For each joint regime, the parameter set θ is characterised by the means (μ^i), standard deviation (σ^i) per unit time and the instantaneous correlation (ρ) of both assets for $i = 1, 2$ such that $\theta = (\mu^1, \mu^2, \sigma^1, \sigma^2, \rho)$. The price of each asset i at time t is then found using the stochastic differential equation

$$dS_t^i = \mu^i S_t^i dt + \sigma^i S_t^i dW_t^i,$$

for $t \geq 0$, where W_t^i are correlated Brownian motions for $i = 1, 2$. The solution to this equation is well-known, and there is a simple method for correlating two Brownian motions such that they have correlation ρ. We are then left with

$$S_t^1 = S_0^1 \exp\left(\mu^1 - \frac{(\sigma^1)^2}{2}\right)t + \sigma^1\sqrt{t}Z^1,$$

$$S_t^2 = S_0^2 \exp\left(\mu^2 - \frac{(\sigma^2)^2}{2}\right)t + \sigma^2\sqrt{t}(\rho Z^1 + \sqrt{(1-\rho^2)}Z^2),$$

where Z^1, Z^2 are standard normal random variables.

In the following experiments we have one normal regime and J alternative regimes, giving us a total of $J + 1$ types of regime. Each regime is characterised by its parameter set θ_j for j in $[0, J]$. When generating the plots shown, we simulated a path for each asset over $T = 20$ market years and each regime change had a duration of $l_i^j = 0.5 \times 252 \times 7$ for $i = 1, \ldots, r^j$ and $j \in [1, J]$. This corresponds to approximately half a year. Our mesh grid was thus

$$\Delta = \left\{ \left[\frac{i-1}{1764}, \frac{i}{1764} \right] : i = 1, 2, \ldots, 252 \times 7 \times 20 \right\}.$$

For each experiment we focused on changing one aspect of the joint distribution of our two assets and evaluated our various algorithms' performance using the metrics in definition 5.0.2. We began by fixing the correlation between the two assets and shifting between a bull and bear market regime characterised by their different mean and variance. We then took the opposite approach and fixed the mean-variance regime while altering the correlation between the assets. Finally we fixed neither and allowed our assets to experience a mixture of both changes in mean and variance, and in correlation.

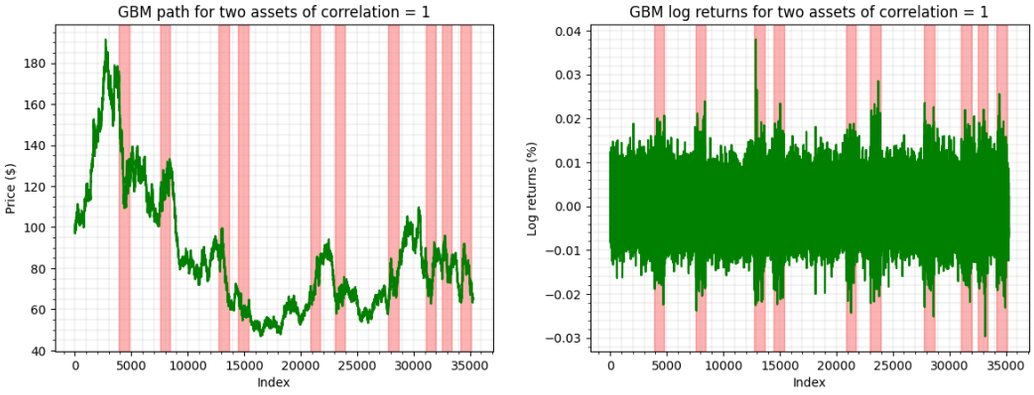
5.1.1 Fixed correlation regime

In this experiment we tested how well the 2-d WK-means and 2-d MMDK-means algorithms picked up a change in the mean-variance regime characterised by a change in the mean and variance of the generator GBM. In order to do so, we fixed the correlation ρ for values of $\rho = -1, -0.5, 0, 0.5$, and 1. We compared their results to those of the univariate algorithms tested on each asset individually using the p-Wasserstein distance for $p = 1, 2$. We generated geometric Brownian motion paths for each asset using the parameter sets

$$\theta_0 = (0.02, 0.02, 0.2, 0.2, \rho), \quad \text{and} \quad \theta_1 = (-0.02, -0.02, 0.3, 0.3, \rho)$$

such that θ_0 and θ_1 correspond to a bull (JR_0) and bear market (JR_1) respectively, and we took $S_0^i = 100$ for $i = 1, 2$.

We began by fixing our assets such that both would undergo a regime change at the same time. Thus we should expect to have two combinations of joint regime JR_0 = (Bull, Bull, Normal) and JR_1 = (Bear, Bear, Normal) and we therefore take $k = 2$. We then fixed $\rho = 1$. When $\rho = 1$ and both assets undergo regime change at the same time, then, in effect, we have created two assets that should in fact move as one. The goal of this experiment was to compare and contrast the performance of the 2-d algorithms with respect to that of the uni-d WK-means. We have only two regimes and hence we have $J = 1$. We took $r^1 = 10$ meaning that approximately a quarter of our path was generated using the JR_1 parameter set with the rest generated from the JR_0 set. Figure 5.3(a) shows an example of such GBM paths with the regime change periods highlighted via a filled in red background colour. Figure 5.3(b) similarly shows the log-returns generated by such price paths.



(a) GBM price paths with $\rho = 1$, regime changes highlighted.

(b) GBM log-returns with $\rho = 1$, regime changes highlighted.

Figure 5.3: GBM synthetic price paths with $\rho = 1$, and their associated log-returns.

We ran the 2-d WK-means and 2-d MMDK-means algorithms, and the uni-d p-WK-means algorithms for $p = 1, 2$ on the same simulated path. We report the accuracy scores for each algorithm in table 5.2 for a total of $n = 50$ runs. We can see from the accuracy scores that the 2-d WK-means algorithm is able to accurately pick up the regime changes' start and end points and, in this regard, all three WK-means algorithms tend to show similar results. The uni-d 2-WK-means algorithm does appear to be superior overall however, with a significantly higher regime-off score. Although the 2-d MMDK-means algorithm returns weaker scores in comparison, these still remain strong with a narrow confidence interval.

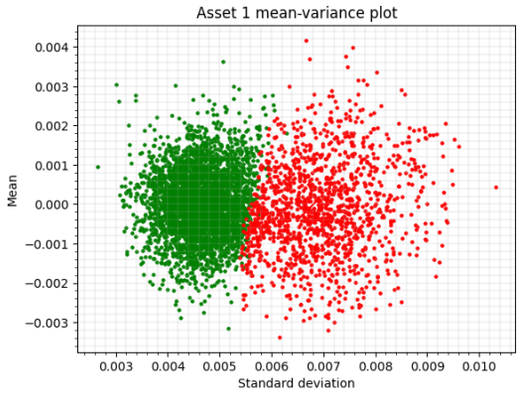
Algorithm	Total	Regime-on	Regime-off
2-d WK-means	85.61% ± 3.54%	87.11% ± 2.78%	85.12% ± 4.42%
Uni-d 1-WK-means	86.46% ± 4.04%	89.45% ± 2.64%	85.26% ± 4.82%
Uni-d 2-WK-means	94.28% ± 1.82%	86.78% ± 2.34%	96.55% ± 1.87%
2-d MMDK-means	77.57% ± 4.10%	81.59% ± 3.86%	76.05% ± 5.62%

Table 5.2: Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = 1$, $n = 50$ runs.

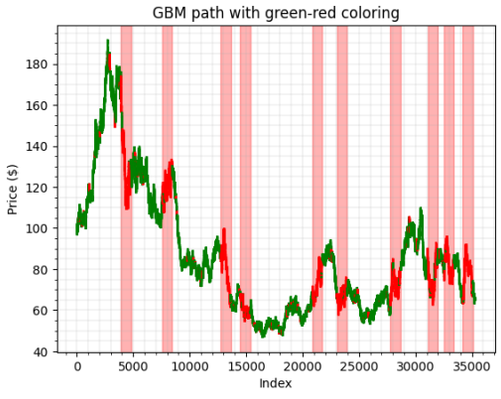
We visualise the clustering generated by the 2-d WK-means using our three types of plot. Given we have $k = 2$, each point is assigned a color, green or red, which signifies which cluster it belongs to. Since $\rho = 1$, both assets will have the same mean-variance plot and so, we need only show that of asset S_1 in figure 5.4(a). We mark with a cross the position of the centroid of each cluster. Upon inspection, we see that the algorithm has done a good job of picking up the two clusters and we note that the red cluster centroid has a higher variance and lower mean than that of the green cluster centroid, as we would expect from our parameter set. Figure 5.4(b), which shows the simulated historical price series for S_1 colored according to each segment's associated cluster, also reinforces the strong detection properties of the 2-d WK-means algorithm.

Finally, figure 5.4(c) shows the correlation plot of the clusters and figure 5.4(d) shows the correlation plot for the atoms of each centroid. The log-returns across both clusters exhibit a correlation of $\rho = 1$, as we would expect, and the atoms of both centroids reflect this as well.

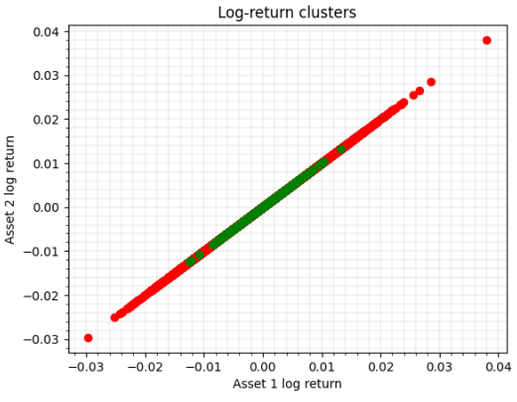
Having tested the case of $\rho = 1$, we then subsequently varied the correlation to values of $\rho = -1, -0.5$, and 0.5. Results for the 2-d WK-means and 2-d MMDK-means algorithms are included in appendix A.3.1 for each experiment. We note that, apart from a particularly strong showing by the 2-d MMDK-means algorithm for $\rho = 0.5$, for each value of ρ the regime-on accuracy score remains relatively stable and strong with only small divergences across the scores. We provide the plots pertaining to $\rho = 0$ in figures 5.5(a) and 5.5(b). Although we see a weaker clustering in the example mean-variance plot for $\rho = 0$ than that of $\rho = 1$, the clustering remains strong, as per the accuracy scores in table 5.3. We provide the accuracy scores for both uni-d algorithms for S_1 as a comparison.



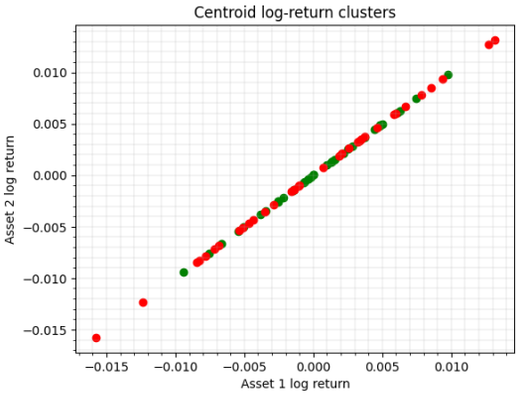
(a) S_1 mean-variance clustering plot.



(b) S_1 price series segmented by cluster association.



(c) Clusters 1 and 2 log-returns plot with cluster colours.

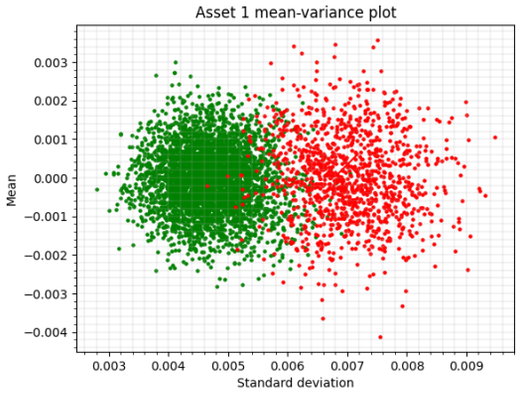


(d) Centroids 1 and 2 log-returns plot with cluster colors.

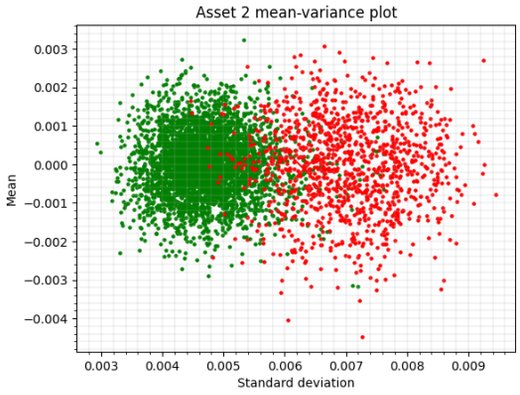
Figure 5.4: GBM, $\rho = 1$, example mean-variance, correlation and price series plots.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	84.36% ± 4.19%	86.52% ± 4.35%	83.45% ± 4.65%
Uni-d 1-WK-means	86.37% ± 3.90%	89.50% ± 3.10%	85.13% ± 4.68%
Uni-d 2-WK-means	94.94% ± 1.55%	89.82% ± 1.21%	96.43% ± 1.66%
2-d MMDK-means	84.63% ± 4.21%	86.57% ± 5.22%	83.78% ± 4.57%

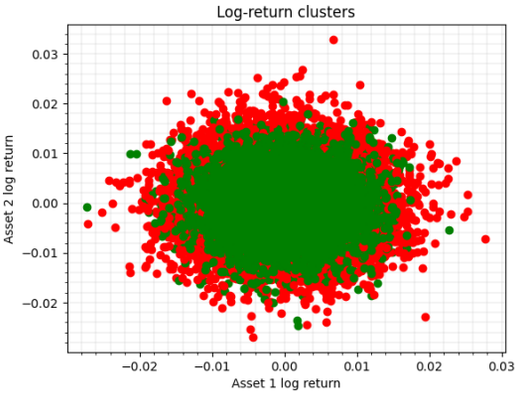
Table 5.3: Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = 0$, $n = 50$ runs.



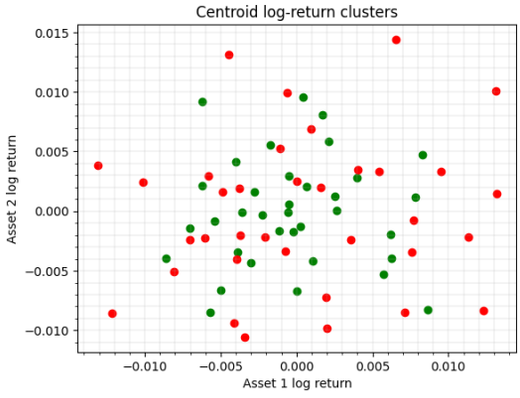
(a) S_1 mean-variance clustering plot.



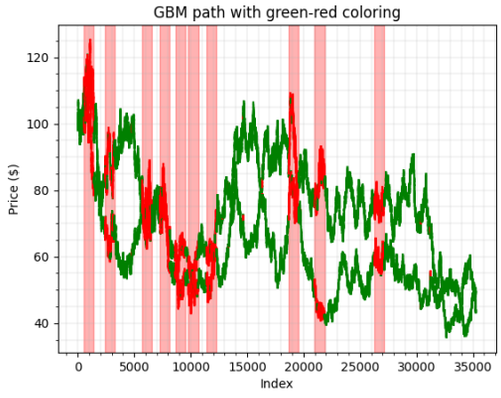
(b) S_2 mean-variance clustering plot.



(c) Clusters 1 and 2 log-returns plot with cluster colours.



(d) Centroids 1 and 2 log-returns plot with cluster colors.



(e) Price series segmented by cluster association.

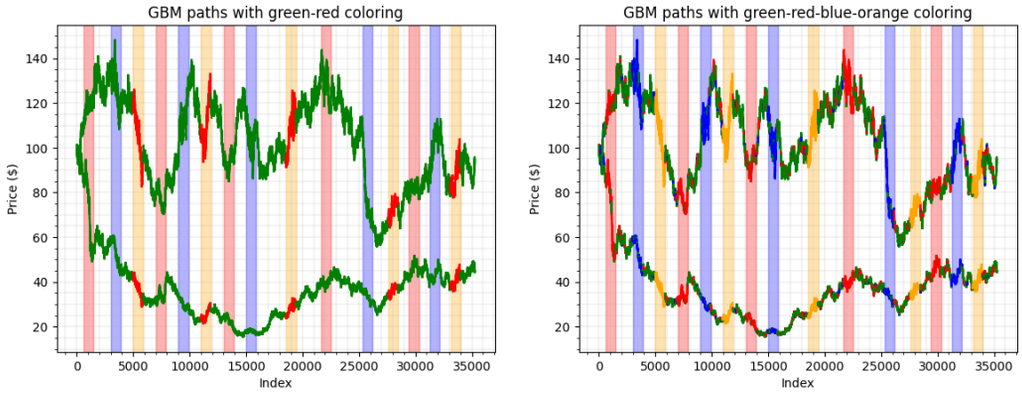
Figure 5.5: GBM, $\rho = 0$, example mean-variance, correlation and price series plots.

Keeping $\rho = 0$, in our next experiment we allowed the mean-variance regimes of each asset to move freely from one another. We therefore had four possible joint regimes: JR_0 = (Bull, Bull, Normal), JR_1 = (Bull, Bear, Normal), JR_2 = (Bear, Bull, Normal), and JR_3 = (Bear, Bear, Normal). We thus had four parameter sets given by

$$\theta_0 = (0.02, 0.02, 0.2, 0.2, \rho), \quad \theta_1 = (0.02, -0.02, 0.2, 0.3, \rho),$$

$$\theta_2 = (-0.02, 0.02, 0.3, 0.2, \rho), \quad \text{and} \quad \theta_3 = (-0.02, -0.02, 0.3, 0.3, \rho),$$

and $J = 3$. We took $r^1 = 5$, $r^2 = 5$ and $r^3 = 5$ meaning that approximately an eighth of our path was generated using each of θ_1, θ_2, and θ_3 and the rest generated from θ_0. As an example of what can go wrong should we choose the wrong number of clusters, taking $k = 2$ yields an inconsistent and poor clustering. Anecdotally, we find that the algorithm tends to focus on one type of regime change, as seen in figure 5.6(a). In this example, time spent in the red section of the plot relates to JR_1, in the blue section relates to JR_2 and in the orange section relates to JR_3. All other points fall under JR_0. We see that the algorithm successfully identifies two market regimes, JR_0 and JR_3 but fails to find the other two. This is to be expected since we are attempting to cluster four market regimes with only two clusters.



(a) GBM price paths with four regimes, $\rho = 0$ and $k = 2$.

(b) GBM price paths with four regimes, $\rho = 0$ and $k = 4$.

Figure 5.6: GBM price paths with four regimes and $\rho = 0$, example plots.

We therefore increased the number of clusters to $k = 4$ and ran the 2-d WK-means and 2-d MMDK-means algorithms. We report the accuracy scores for each algorithm in table 5.4 for a total of $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	62.94% ± 4.01%	71.58% ± 5.69%	57.60% ± 6.30%
2-d MMDK-means	42.54% ± 2.54%	56.35% ± 5.41%	34.16% ± 3.28%

Table 5.4: Accuracy scores with 95% CI, GBM synthetic path with four different mean-variance regimes and fixed $\rho = 0$, $n = 50$ runs.

Clearly there is a marked drop in the accuracy scores when compared to previous cases. The regime-on score for the 2-d WK-means algorithm still remains relatively strong but this comes at the expense of the regime-off score which suffers greatly. In comparison, each of the accuracy scores for the 2-d MMDK-means algorithm are significantly impacted. Figure 5.6(b) shows an example path generated using $k = 4$ clusters and such that the path is segmented into four colors: green, red, blue and orange.

5.1.2 Fixed mean-variance regime

In this series of experiments we tested how well the 2-d algorithms picked up a change in correlation regime, characterised by a change in the correlation ρ between the two assets. In order to do so, we fixed the mean-variance regime and varied the correlation between two values ρ_0 and ρ_1. We generated geometric Brownian motion paths for each asset using the parameter sets

$$\theta_0 = (0.02, 0.02, 0.2, 0.2, \rho_0), \quad \text{and} \quad \theta_1 = (0.02, 0.02, 0.2, 0.2, \rho_1)$$

such that θ_0 and θ_1 correspond to a normal and abnormal correlation regime respectively. Again we took $S_0^i = 100$ for $i = 1, 2$. In these experiments we always had exactly two different correlation regimes and thus we took $k = 2$ throughout. Therefore $J = 1$ and we took $r^1 = 10$, meaning that approximately a quarter of our path was generated using θ_1, and the rest was generated from θ_0. Formally, our regimes are JR_0 = (Bull, Bull, Normal), and JR_1 = (Bull, Bull, Abnormal).

We began by fixing $\rho = (\rho_0, \rho_1) = (0, 1)$. Our assets therefore have a correlation of 0 three quarters of the time but a correlation of 1 otherwise. Such a jump in correlation initially appears extreme but during times of crisis it is well-known that the correlation between assets often increases significantly [74]. We report the accuracy scores for the 2-d WK-means and 2-d MMDK-means algorithms in table 5.5 for a total of $n = 50$ runs. We see from the accuracy scores that both algorithms perform very well in picking up the regime changes.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	92.71% ± 4.72%	98.98% ± 0.14%	90.40% ± 6.29%
2-d MMDK-means	99.46% ± 3.76%	99.20% ± 0.00%	99.32% ± 0.00%

Table 5.5: Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = 1$, $n = 50$ runs.

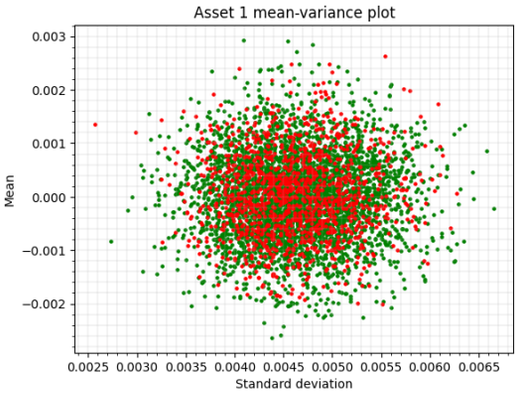
We visualise the clustering using our three types of plot. Figures 5.7(a) and 5.7(b) show our mean-variance plot for each asset, S_1 and S_2. As we would expect, there are no discernible clusters when plotting in mean-variance space due to the fixed mean-variance regime. In contrast, our correlation plots for each cluster's log-returns in figure 5.7(c) and each centroid's log-returns in 5.7(d) show that our algorithm has found clear relationships in the data. The second cluster's log-returns and the associated centroid's atoms clearly fall on the $x = y$ axis while those of the first cluster are spread out evenly around the centre. Finally, figure 5.7(e) gives a visual demonstration of the strength of our algorithms in picking up changes in correlation through time.

Having tested the case of $\rho = (0, 1)$, we then subsequently varied the values of ρ_0 and ρ_1 between -1 and 1. Complete results are included in appendix A.3.1 for each experiment. We find that both algorithms report strong accuracy scores for the cases of $\rho = (1, 0)$ $\rho = (0, -1)$ and $\rho = (-1, 0)$. This was reassuring as it indicated that the algorithm does not respond uniquely to the case of $\rho = (0, 1)$ but instead picks up the structural changes in the correlation value. It is noticeable however, that the regime-on accuracy score does tend to be somewhat weaker for the cases of $\rho = (1, 0)$ and $\rho = (-1, 0)$ when compared to $\rho = (0, 1)$ and $\rho = (0, -1)$.

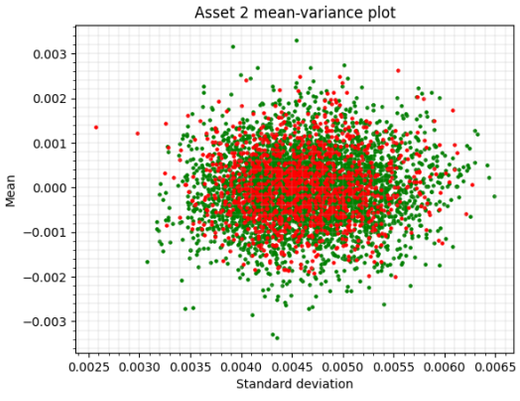
One might reasonably question whether the severity of the correlation changes impacts the accuracy of the algorithm. In order to test this, we took $\rho = (0.5, 1)$, $\rho = (-0.5, -1)$, and $\rho = (0, 0.5)$. We report the accuracy scores for $\rho = (0.5, 1)$ in table 5.6 for a total of $n = 50$ runs. Although there is a slight drop in each respective score when compared to $\rho = (0, 1)$, overall the algorithms still perform strongly. Results for $\rho = (-0.5, -1)$ and $\rho = (0, 0.5)$ are included in appendix A.3.1.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	87.78% ± 5.21%	91.85% ± 4.76%	86.22% ± 6.09%
2-d MMDK-means	97.41% ± 2.78%	96.56% ± 3.22%	97.33% ± 2.81%

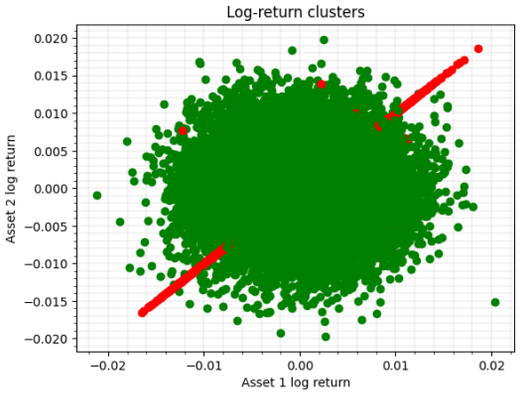
Table 5.6: Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 0.5$, $\rho_1 = 1$, $n = 50$ runs.



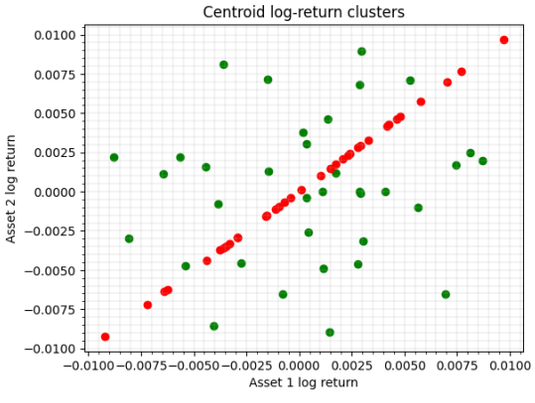
(a) S_1 mean-variance clustering plot.



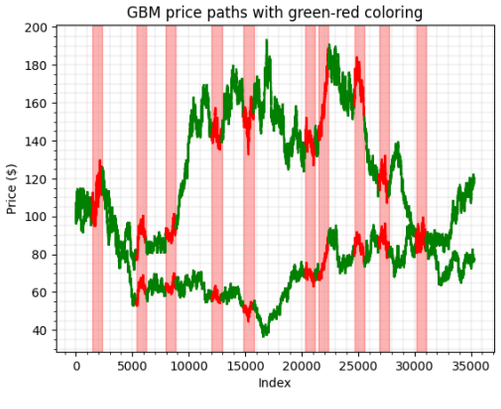
(b) S_2 mean-variance clustering plot.



(c) Clusters 1 and 2 log-returns plot with cluster colours.

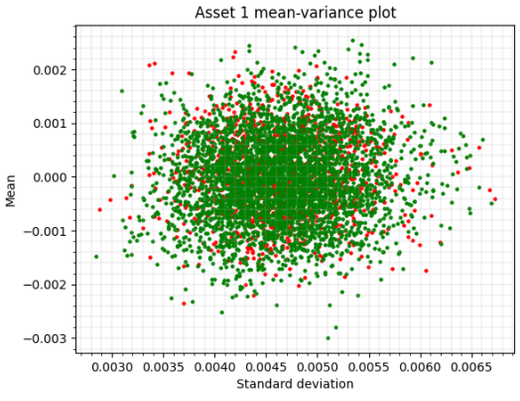


(d) Centroids 1 and 2 log-returns plot with cluster colors.

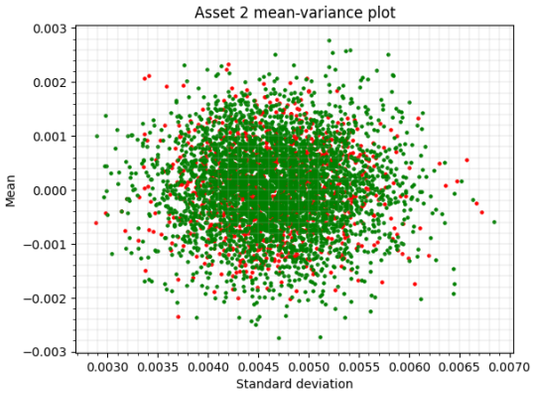


(e) Price series segmented by cluster association.

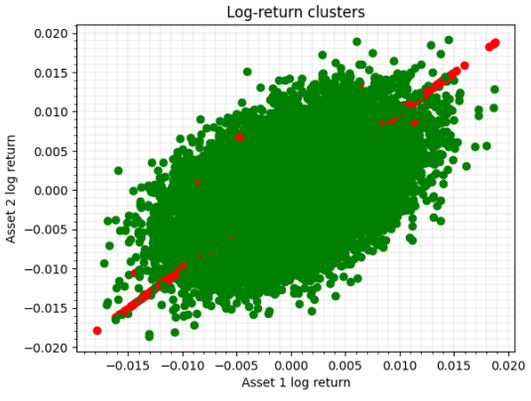
Figure 5.7: GBM with correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, example mean-variance, correlation and price series plots.



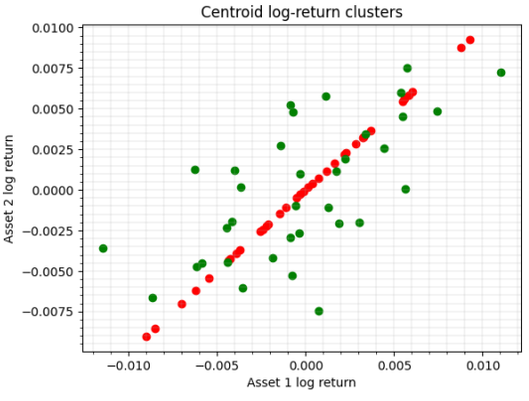
(a) S_1 mean-variance clustering plot.



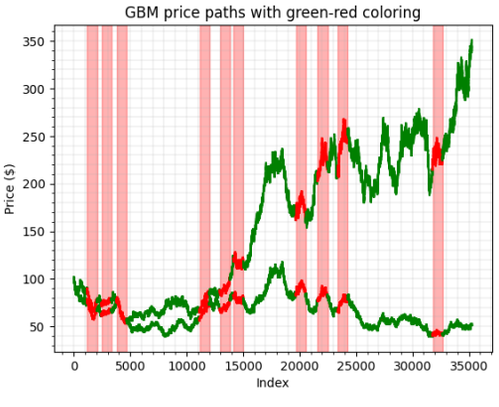
(b) S_2 mean-variance clustering plot.



(c) Clusters 1 and 2 log-returns plot with cluster colours.



(d) Centroids 1 and 2 log-returns plot with cluster colors.



(e) Price series segmented by cluster association.

Figure 5.8: GBM with correlation regimes, $\rho_0 = 0.5$ and $\rho_1 = 1$, example mean-variance, correlation and price series plots.

Figure 5.8 includes the three plots associated with $\rho = (0.5, 1)$. Again we see similar results to before and we may reasonably conclude that the algorithms tend to pick up changes in the correlation regime change, severe or otherwise. One trend that did appear is that the clustering tends to be weaker for jumps of 0.5 in correlation when the normal correlation is 0, as shown in the results for $\rho = (0, 0.5)$ in the appendix. In such pairings, the 2-d WK-means algorithm does well in picking up the correlation regime changes but returns weaker scores than in other pairings of (ρ_0, ρ_1). In contrast the 2-d MMDK-means algorithm struggles considerably and returns a much lower regime-on accuracy score. It is also noticeable across most tests that the 2-d WK-means algorithm tends to respond more strongly to changes in correlation regime than the 2-d MMDK-means algorithm.

5.1.3 Free correlation and mean-variance regime

In our final set of experiments, we varied both the mean-variance regime and correlation regime. To begin, we tested three joint market regimes: JR_0 = (Bull, Bull, Normal), JR_1 = (Bull, Bull, Abnormal) and JR_2 = (Bear, Bear, Normal). We generated geometric Brownian motion paths for each asset using the parameter sets

$$\theta_0 = (0.02, 0.02, 0.2, 0.2, 0), \quad \theta_1 = (0.02, 0.02, 0.2, 0.2, 1),$$

$$\text{and} \quad \theta_2 = (-0.02, -0.02, 0.3, 0.3, 0).$$

We took $S_0^i = 100$ for $i = 1, 2$, $J = 2$ and $r^1 = r^2 = 5$. We ran the 2-d algorithms for $k = 3$ clusters. We report the accuracy scores for each algorithm in table 5.7 for a total of $n = 50$ runs.

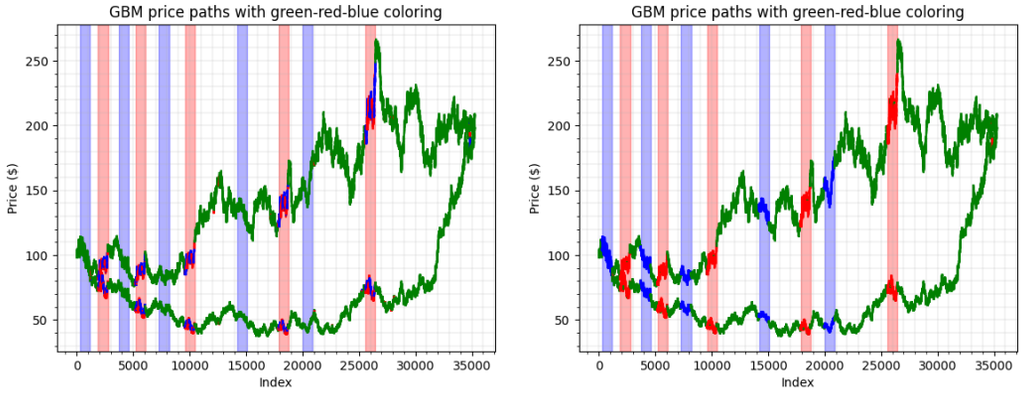
Algorithm	Total	Regime-on	Regime-off
2-d WK-means	70.10% ± 6.18%	61.78% ± 10.24%	72.70% ± 5.87%
2-d MMDK-means	70.83% ± 5.58%	71.99% ± 10.54%	70.28% ± 4.73%

Table 5.7: Accuracy scores with 95% CI, GBM synthetic path, $k = 3$, with correlation and mean-variance regimes, $n = 50$ runs.

Algorithm	Regime-on JR_1	Regime-on JR_2
2-d WK-means	56.35% ± 10.15%	67.20% ± 10.15%
2-d MMDK-means	65.85% ± 12.40%	78.12% ± 10.26%

Table 5.8: Accuracy scores with 95% CI, GBM synthetic path, $k = 3$, with correlation (JR_1) and mean-variance (JR_2) regimes, $n = 50$ runs.

We note the drop in accuracy for both algorithms as seen in table 5.7 when changing both the mean-variance and correlation regime compared to changing only one. The 2-d WK-means algorithm struggles to cluster for both regimes and often picks up one whilst ignoring the other as exemplified by figure 5.9(a). This leads to a large variation in the regime-on score for each type of regime as demonstrated in table 5.8. Figure 5.9(b) is an example of a successful clustering where each type and period of regime change has been identified. Green segments are those generated by JR_0, red are those generated by JR_1 and blue are those generated by JR_2. The 2-d MMDK-means algorithm outperforms the 2-d WK-means algorithm and still retains relatively strong accuracy scores. We note that on this occasion, both algorithms have done a better job of picking up changes in the mean-variance regime than in the correlation regime as shown in table 5.8.



(a) Price series segmented by cluster association, incorrect clustering.

(b) Price series segmented by cluster association, correct clustering.

Figure 5.9: GBM price paths with three market and correlation regimes that do not overlap segmented by cluster association, $\rho_0 = 0$ and $\rho_1 = 1$, example plots.

We then tested four joint market regimes: JR_0 = (Bull, Bull, Normal), JR_1 = (Bull, Bull, Abnormal), JR_2 = (Bear, Bear, Normal) and JR_3 = (Bear, Bear, Abnormal). We generated geometric Brownian motion paths for each asset using the parameter sets

$$\theta_0 = (0.02, 0.02, 0.2, 0.2, 0), \quad \theta_1 = (0.02, 0.02, 0.2, 0.2, 1),$$

$$\theta_2 = (-0.02, -0.02, 0.3, 0.3, 0), \quad \text{and} \quad \theta_3 = (-0.02, -0.02, 0.3, 0.3, 1).$$

We took $S_0^i = 100$ for $i = 1, 2$, $J = 3$ and $r^1 = r^2 = r^3 = 5$. We then ran the 2-d algorithms for $k = 4$ clusters. We report the accuracy scores for each algorithm in table 5.9 for a total of $n = 50$ runs.

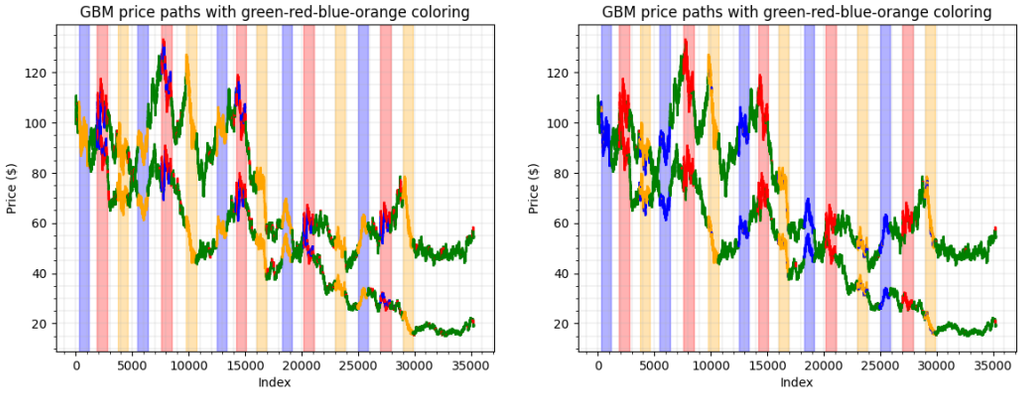
Algorithm	Total	Regime-on	Regime-off
2-d WK-means	63.52% ± 5.04%	61.46% ± 5.10%	64.57% ± 6.41%
2-d MMDK-means	60.49% ± 3.08%	61.85% ± 3.59%	59.51% ± 4.59%

Table 5.9: Accuracy scores with 95% CI, GBM synthetic path, $k = 4$, with correlation and mean-variance regimes, $n = 50$ runs.

Algorithm	Regime-on JR_1	Regime-on JR_2	Regime-on JR_3
2-d WK-means	58.77% ± 9.08%	43.57% ± 13.09%	82.03% ± 8.88%
2-d MMDK-means	21.87% ± 10.14%	70.01% ± 9.77%	93.67% ± 3.85%

Table 5.10: Accuracy scores with 95% CI, GBM synthetic path, $k = 3$, with correlation (JR_1), mean-variance (JR_2) and joint correlation-mean-variance (JR_3) regimes, $n = 50$ runs.

Our accuracy scores in table 5.10 make for interesting reading. The algorithms again struggle to cluster for both regimes where one of the mean-variance regime or correlation regime changes (JR_1 and JR_2 respectively), with quite a wide confidence interval and low score. However, both the 2-d WK-means and 2-d MMDK-means algorithms are far better at picking up instances of change when both the mean-variance and correlation regimes change together (JR_3), returning accuracy scores of above 80% and 90% respectively. It is also noticeable that the 2-d WK-means algorithm was better at picking up changes in the correlation regime whilst the opposite was true for the 2-d MMDK-means algorithm, which displayed stronger scores when picking up changes in the mean-variance regime. The overall scores remain low as seen in table 5.9. Inspection of the corresponding price series with segmented colors often helps us to ascertain whether the clustering has been effective. Figures 5.10(a) and 5.10(b) are examples of such plots. Green segments are those generated by JR_0, red are those generated by JR_1, blue are those generated by JR_2 and orange are those generated by JR_3.



(a) Price series segmented by cluster association, incorrect clustering.

(b) Price series segmented by cluster association, correct clustering.

Figure 5.10: GBM price paths with four mean-variance and correlation regimes segmented by cluster association, $\rho_0 = 0$ and $\rho_1 = 1$, example plots.

5.2 Merton jump diffusion process

In order to further test our algorithms, we apply them to a non-Gaussian environment. In this section we generate two synthetic price series modelled as Merton jump diffusion processes. In the uni-dimensional case, the stock price of one asset is then the solution to the stochastic differential equation

$$dS_t^i = \mu^i S_t^i dt + \sigma^i S_t^i dW_t^i + S_{t-}^i dJ_t^i,$$

for $t \geq 0$, where (W_t^1, W_t^2) are correlated Brownian motions and

$$J_t^i := \sum_{j=1}^{N_t} Y_j^i - 1,$$

for $i = 1, 2$. Here, we have that $N_t \sim \text{Po}(\lambda t)$ is a Poisson random variable, and $\ln(1 + Y_j) \sim \text{Normal}(\gamma^i, (\delta^i)^2)$. We note that this equation, and indeed solution, is similar to the previous geometric Brownian motion case but with an added jump component.

When working with two assets, it is not sufficient to simply correlate the Brownian motions alone however. We must also correlate the jumps generated in each process. As in [75], we model each price series such that the j^{th} jumps Y_j^1 and Y_j^2 are correlated with instantaneous correlation ρ and occur together, driven by the same Poisson arrival process N_t. The case of one jump size being zero and one being non-zero is possible, thus allowing our assets to experience idiosyncratic shocks along with joint shocks. A full description of the process and results are described in appendix A.3.2. This model yields a solution of the form

$$S_t^i = S_0^i \exp\left[\left(\mu^i - \lambda\kappa^i - \frac{(\sigma^i)^2}{2}\right)t + \sigma^i\sqrt{t}Z^i + \sum_{j=1}^{N_t} Y_j^i\right],$$

with

$$\kappa^i := \mathbb{E}[e^{Y^i} - 1],$$

for $i = 1, 2$ and where Z^1, Z^2 are standard normal random variables and Y^1, Y^2 are correlated normal random variables such that $Y^i \sim \text{Normal}(\gamma^i, (\delta^i)^2)$.

Again, we take $\mathcal{M}(\theta)$ to be a family of models indexed by a parameter set $\theta \subset \mathbb{R}^{10}$. For each regime, the parameter set θ is characterised by the means (μ^i) and standard deviations (σ^i) per unit time of the assets used to generate the geometric Brownian motion, the intensity (λ) of the Poisson process, the means (γ^i) and standard deviations (δ^i) per unit time of the jump process, and the instantaneous correlation (ρ) of both assets for $i = 1, 2$ such that $\theta = (\mu^1, \mu^2, \sigma^1, \sigma^2, \lambda, \gamma^1, \gamma^2, \delta^1, \delta^2, \rho)$.

We apply the same methodology in this section as we did in section 5.1. In the following experiments we have one normal regime and J different regimes, giving us $J+1$ types of regime. Each regime is characterised by its parameter set θ_j for j in $[0, J]$. We simulate a path using the correlated Merton jump diffusion process for each asset over $T = 20$ years and each regime change will have a duration of $l_i^j = 0.5 \times 252 \times 7$ for $i = 1, \ldots, r^j$ and $j \in [1, J]$.

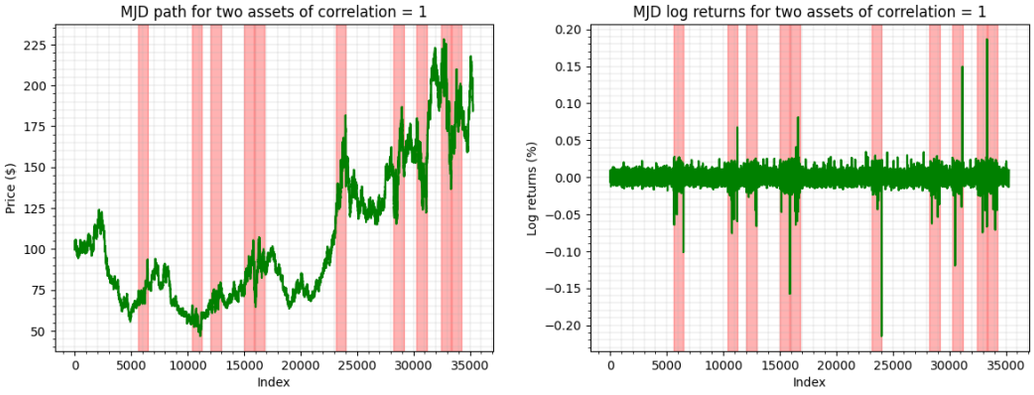
5.2.1 Fixed correlation regime

In this experiment we tested how well the 2-d algorithms picked up a change in mean-variance regime characterised by a change in the mean and variance of the generator MJD. The testing structure is analogous to that of section 5.1.1. We fixed the correlation ρ for values of $\rho = -1, -0.5, 0, 0.5$, and 1 and compared the results of our novel algorithms to those of the univariate algorithm tested on each asset individually using the 1- and 2-Wasserstein distances. We generated Merton jump diffusion paths for each asset using the parameter sets

$$\theta_0 = (0.05, 0.05, 0.2, 0.2, 5, 0.02, 0.02, 0.005, 0.005, \rho), \quad \text{and}$$
$$\theta_1 = (-0.05, -0.05, 0.35, 0.35, 10, -0.04, -0.04, 0.04, 0.04, \rho)$$

such that θ_0 (JR_0) and θ_1 (JR_1) correspond to a bull and bear market regime respectively. Once again, we took $S_0^i = 100$ for $i = 1, 2$.

We began by fixing our assets such that both would undergo a regime change at the same time. Thus we have two combinations of joint regime JR_0 = (Bull, Bull, Normal) and JR_1 = (Bear, Bear, Normal) and take $k = 2$. We then fixed $\rho = 1$. We have only two regimes and hence we took $J = 1$. We took $r^1 = 10$ meaning that approximately a quarter of our path was generated using the JR_1 parameter set with the rest generated from the JR_0 set. Figure 5.11(a) shows an example of such a MJD path with the regime change periods highlighted on the grid in red. Alongside, figure 5.11(b) shows the log-returns associated to this example path, again with the regime changes highlighted.



(a) MJD price paths with $\rho = 1$, regime changes highlighted.

(b) MJD log-returns with $\rho = 1$, regime changes highlighted.

Figure 5.11: MJD synthetic price paths with $\rho = 1$, and their associated log-returns.

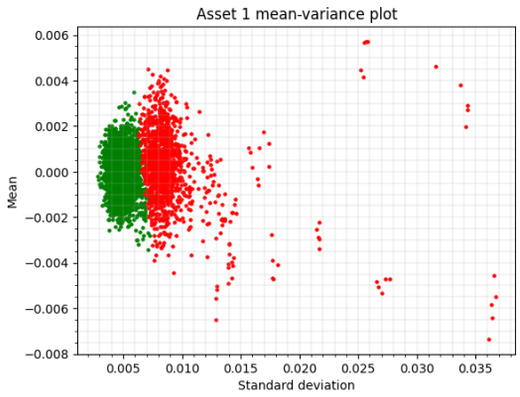
We ran the algorithms on the same simulated path. We report the accuracy scores for each algorithm in table 5.11 for a total of $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	94.04% ± 1.90%	81.46% ± 8.05%	98.00% ± 0.15%
Uni-d 1-WK-means	93.06% ± 2.45%	85.37% ± 8.38%	95.40% ± 2.33%
Uni-d 2-WK-means	94.63% ± 1.88%	81.80% ± 7.73%	98.68% ± 0.07%
2-d MMDK-means	90.82% ± 2.12%	93.97% ± 1.00%	89.56% ± 3.00%

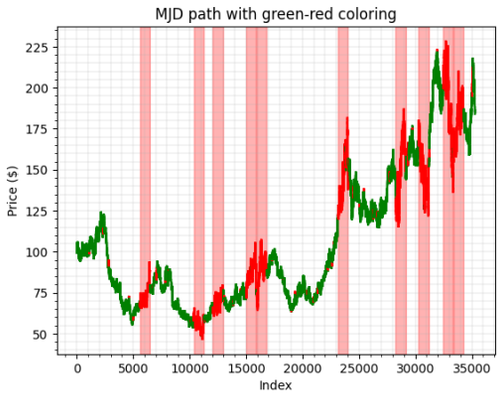
Table 5.11: Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and $\rho = 1$, $n = 50$ runs.

The 2-d WK-means, uni-d 1- and 2-WK-means algorithms exhibit lower regime-on accuracy scores when compared to the geometric Brownian motion case in 5.1.1 with both the 2-d WK-means and uni-d 2-WK-means algorithms suffering larger drops than the uni-d 1-WK-means algorithm. This is not unreasonable given both the 2-d WK-means and the uni-d 2-WK-means algorithms use the 2-Wasserstein distance and thus may be more susceptible to extreme outliers that would occur in a Merton jump diffusion process. In comparison, the 2-d MMDK-means algorithm returns significantly stronger accuracy scores than its analogous GBM case.

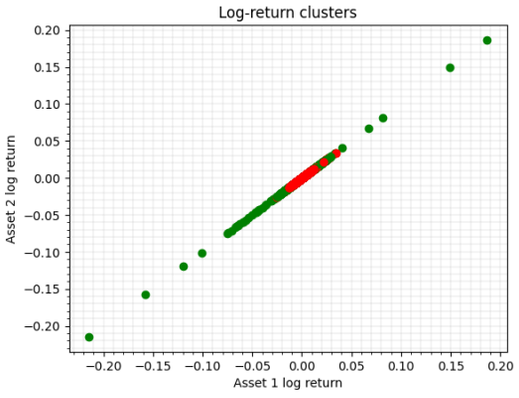
We visualise the clustering generated by the 2-d WK-means using our three types of plot in figure 5.12, analogously to section 5.1.1. As in the GBM case, we see that the algorithm has done a good job of picking up the two clusters and we note that the red cluster centroid has higher variance than that of the green cluster centroid, as we would expect from our parameter set. Figure 5.12(b), which shows the simulated historical price series for S_1 coloured according to each segment's associated cluster, also reinforces the strong detection properties of our algorithms. As expected, the correlation amongst all values in both clusters is approximately one as seen in figures 5.12(c) and 5.12(d).



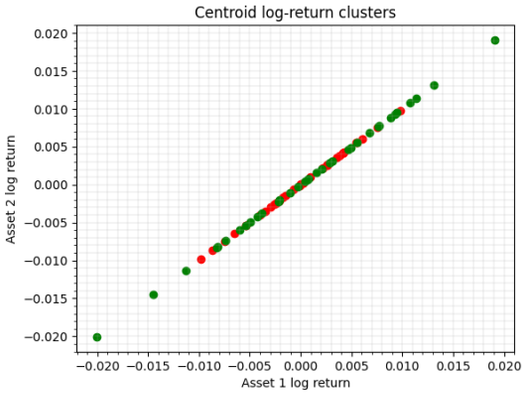
(a) S_1 mean-variance clustering plot.



(b) S_1 price series segmented by cluster association.



(c) Clusters 1 and 2 log-returns plot with cluster colours.



(d) Centroids 1 and 2 log-returns plot with cluster colors.

Figure 5.12: MJD, $\rho = 1$, example mean-variance, correlation and price series plots.

Having tested the case of $\rho = 1$, we then subsequently varied the correlation to values of $\rho = -1, -0.5$, and 0.5 with results for each test included in appendix A.3.3. The accuracy scores for the 2-d WK-means algorithm tend to be lower than the analogous GBM case. The 2-d MMDK-means algorithm, on the other hand, significantly outperforms the 2-d WK-means algorithm, returning very strong scores in each test. The accuracy scores for $\rho = 0$ can be found in table 5.12 along with the related plots in figure 5.13.

Algorithm	Asset	Total	Regime-on	Regime-off
2-d WK-means	-	85.47% ± 4.83%	79.16% ± 9.73%	87.37% ± 5.30%
Uni-d 1-WK-means	1	90.07% ± 3.79%	83.75% ± 8.38%	91.96% ± 4.67%
Uni-d 2-WK-means	1	94.71% ± 1.65%	81.45% ± 6.77%	98.91% ± 0.05%
Uni-d 1-WK-means	2	89.74% ± 3.23%	80.33% ± 9.47%	92.67% ± 3.63%
Uni-d 2-WK-means	2	89.50% ± 2.35%	60.12% ± 9.65%	99.07% ± 0.08%
2-d MMDK-means	-	83.07% ± 7.44%	75.06% ± 11.07%	85.55% ± 6.30%

Table 5.12: Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and $\rho = 0$, $n = 50$ runs.

We focus on the regime-on accuracy score for each algorithm in table 5.12. We note that compared to the case of $\rho = 1$, the accuracy scores of the 2-d WK-means and uni-d 1-WK-means algorithms are relatively similar with only a small reduction. A much more noticeable drop is to be found in the regime-on score for the uni-d 2-WK-means algorithm when applied to S_2. Examining the mean-variance plots in figures 5.13(a) and 5.13(b), we see that S_2 exhibits more extreme jumps than S_1. Given the uni-d 2-WK-means algorithm utilises the mean average when calculating its centroids, as shown in proposition 4.1.6, its performance is therefore more susceptible to extreme outliers and this is reflected in the contrasting regime-on accuracy scores when it is applied to S_1 and S_2. We might then reasonably expect the performance of the 2-d WK-means algorithm to also suffer since both use the 2-Wasserstein distance. In fact its regime-on accuracy score is relatively unchanged, indicating that its method of clustering both assets has made it more robust when compared to its uni-dimensional counterpart.

Overall when testing, we found that an increase in extreme outliers tended to reduce the accuracy scores for all four algorithms but that this was particularly detrimental to the performance of the uni-d 2-WK-means algorithm.

Remark 5.2.1. We note in passing that, when compared to our real data tests in chapter 6, the MJD process used had a tendency to produce more extreme variance outliers which may be seen when comparing mean-variance clustering plots.

Keeping $\rho = 0$, in our next experiment we allowed the mean-variance regimes of each asset to move freely from one another. We therefore had four possible joint regimes: JR_0 = (Bull, Bull, Normal), JR_1 = (Bull, Bear, Normal), JR_2 = (Bear, Bull, Normal), and JR_3 = (Bear, Bear, Normal). Our parameter sets are given by

$$\theta_0 = (0.05, 0.05, 0.2, 0.2, 5, 0.02, 0.02, 0.005, 0.005, \rho),$$

$$\theta_1 = (0.05, -0.05, 0.2, 0.35, 7, 0.02, -0.04, 0.005, 0.04, \rho)$$

$$\theta_2 = (-0.05, 0.05, 0.35, 0.2, 7, -0.04, 0.02, 0.04, 0.005, \rho), \quad \text{and}$$

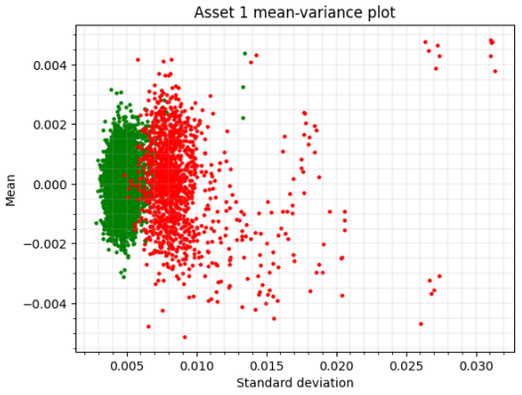
$$\theta_3 = (-0.05, -0.05, 0.35, 0.35, 10, -0.04, -0.04, 0.04, 0.04, \rho),$$

and $J = 3$. We took $r^1 = 5$, $r^2 = 5$ and $r^3 = 5$. In our example plot 5.14(a), taking $k = 2$ yields a good clustering when using the 2-d WK-means algorithm in that our algorithm can pick up the various regime changes. However, it can not distinguish between the different types of regime change and thus all types are classified as one.

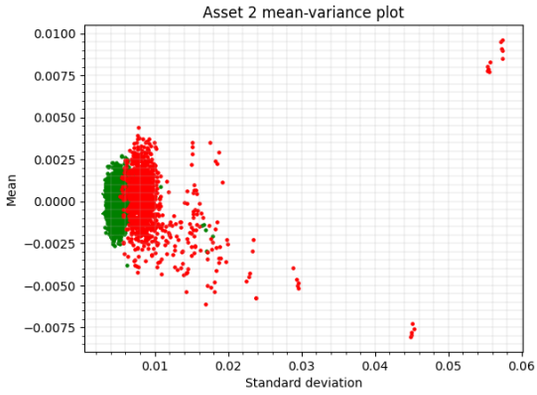
We therefore increased the number of clusters to $k = 4$ and ran the 2-d algorithms. We report the accuracy scores for each algorithm in table 5.13 for a total of $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	67.19% ± 3.93%	46.91% ± 6.76%	79.14% ± 5.29%
2-d MMDK-means	53.01% ± 4.46%	49.94% ± 4.97%	54.70% ± 5.25%

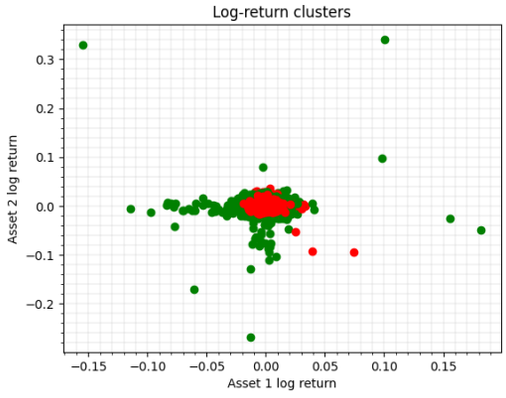
Table 5.13: Accuracy scores with 95% CI, MJD synthetic path with four different mean-variance regimes and $\rho = 0$, $n = 50$ runs.



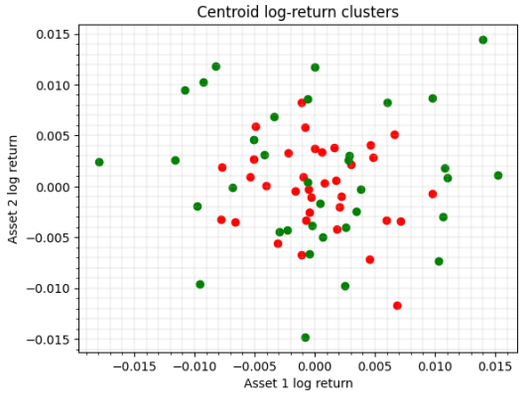
(a) S_1 mean-variance clustering plot.



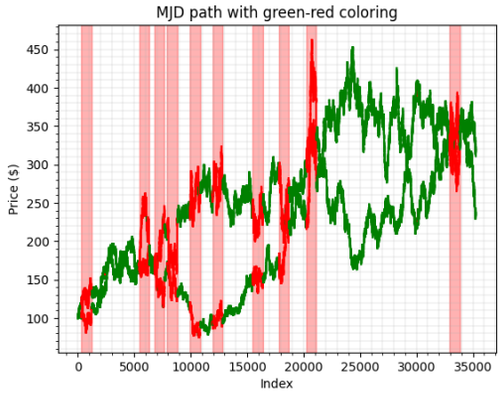
(b) S_2 mean-variance clustering plot.



(c) Clusters 1 and 2 log-returns plot with cluster colours.

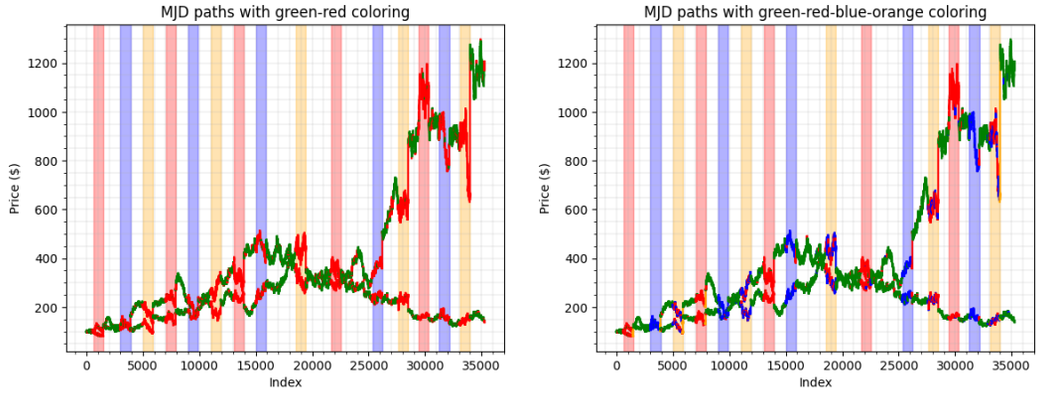


(d) Centroids 1 and 2 log-returns plot with cluster colors.



(e) Price series segmented by cluster association.

Figure 5.13: MJD, $\rho = 0$, example mean-variance, correlation and price series plots.



(a) MJD price paths with four regimes, $\rho = 0$ and $k = 2$.

(b) MJD price paths with four regimes, $\rho = 0$ and $k = 4$.

Figure 5.14: MJD price paths with four regimes and $\rho = 0$, example plots.

Clearly there is a significant drop in the regime-on accuracy scores when compared to previous cases including that of its GBM counterpart. Figure 5.14(b) shows an example path generated using $k = 4$ clusters and such that the path is segmented into four colors: green, red, blue and orange.

5.2.2 Fixed mean-variance regime

In this series of experiments we tested how well the 2-d algorithms picked up a change in correlation regime, characterised by a change in the correlation ρ between the two assets. In order to do so, we fixed the mean-variance regime and varied the correlation between two values ρ_0 and ρ_1. We generated Merton jump diffusion paths for each asset using the parameter sets

$$\theta_0 = (0.05, 0.05, 0.2, 0.2, 5, 0.02, 0.02, 0.005, 0.005, \rho_0), \quad \text{and}$$

$$\theta_1 = (0.05, 0.05, 0.2, 0.2, 5, 0.02, 0.02, 0.005, 0.005, \rho_1)$$

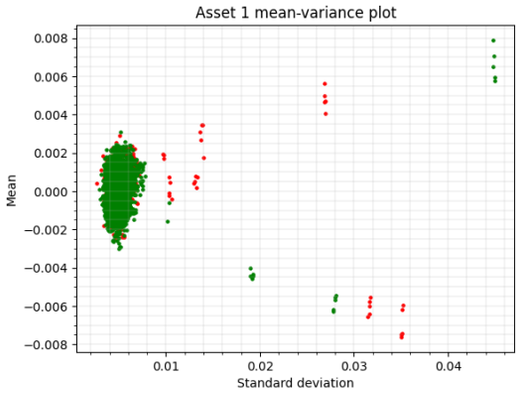
for the normal and abnormal regimes respectively, and once again we took $S_0^i = 100$ for $i = 1, 2$. In these experiments we always had exactly two different correlation regimes and thus we took $k = 2$ throughout. Formally, our regimes are JR_0 = (Bull, Bull, Normal), and JR_1 = (Bull, Bull, Abnormal).

We report the accuracy scores for $\rho = (0, 1)$ in table 5.14 for a total of $n = 50$ runs. As in the GBM case, both 2-d algorithms perform very well in picking up the regime changes.

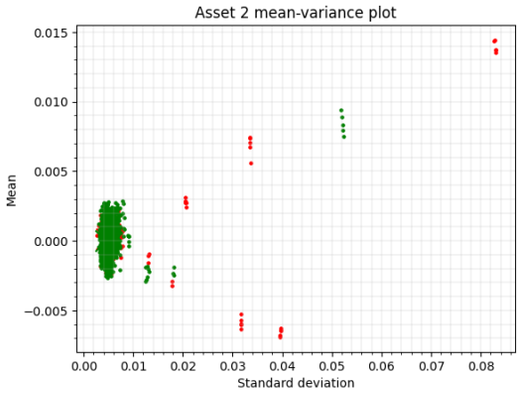
Algorithm	Total	Regime-on	Regime-off
2-d WK-means	95.72% ± 3.35%	92.02% ± 6.30%	96.73% ± 3.69%
2-d MMDK-means	96.02% ± 2.87%	96.91% ± 2.11%	95.50% ± 3.26%

Table 5.14: Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = 1$, $n = 50$ runs.

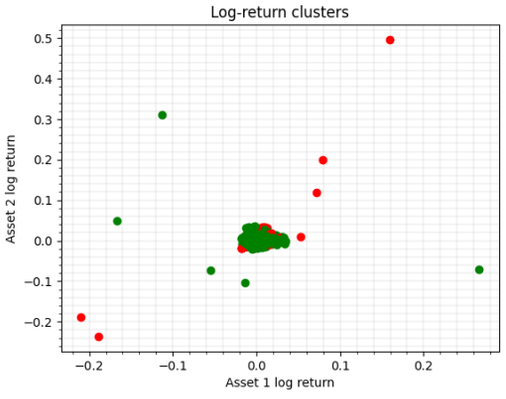
We visualise the clustering generated by the 2-d WK-means. Our correlation plots for each cluster's log-returns in figure 5.15(d) and each centroid's log-returns in 5.15(e) show that the algorithm has found the expected relationships in the data. Although somewhat less clear than the GBM case, the second cluster's log-returns and associated centroid's atoms fall on the $x = y$ axis while those of the first cluster are spread out evenly around the centre. Figure 5.15(f) gives a visual demonstration of the strength of our algorithm in picking up changes in correlation through time.



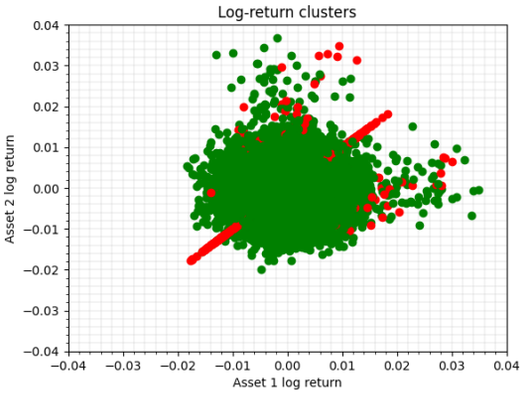
(a) S_1 mean-variance clustering plot.



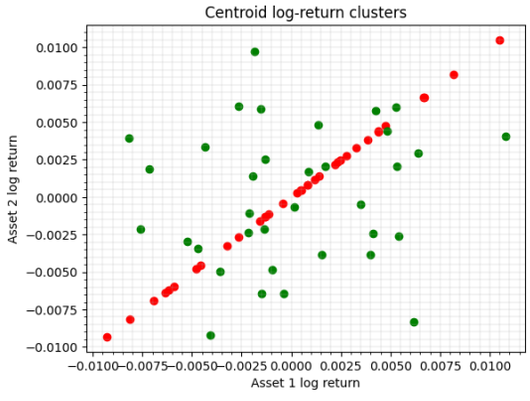
(b) S_2 mean-variance clustering plot.



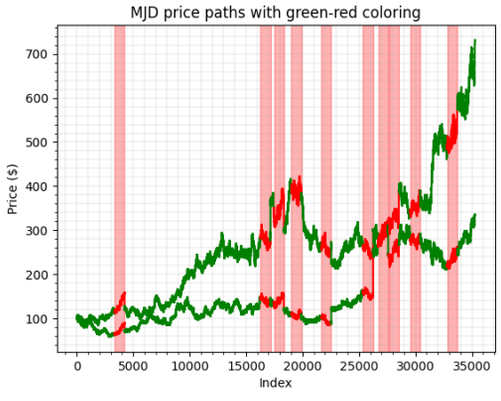
(c) Clusters 1 and 2 log-returns plot with cluster colours.



(d) Clusters 1 and 2 log-returns plot with cluster colours, magnified.



(e) Centroids 1 and 2 log-returns plot with cluster colors.



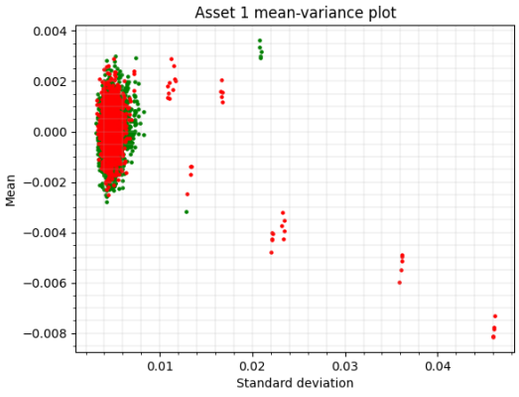
(f) Price series segmented by cluster association.

Figure 5.15: MJD with correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, example mean-variance, correlation and price series plots.

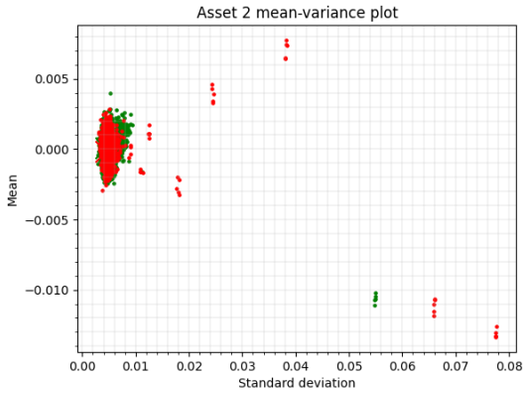
Having tested the case of $\rho = (0, 1)$, we then subsequently varied the values of ρ_0 and ρ_1 between -1 and 1. Again, results are included for each supplementary test in appendix A.3.3. The general trends noted in the GBM case were observed here as well, albeit with each regime-on accuracy score being lower. Taking $\rho = (0.5, 1)$ we observed similar results to the case of $\rho = (0, 1)$ for the 2-d WK-means algorithm. The 2-d MMDK-means algorithm, however, showed a marked decrease in regime-on accuracy score. We report the accuracy scores for each algorithm in table 5.15 for a total of $n = 50$ runs. Figure 5.16 includes the three plots associated with $\rho = (0.5, 1)$.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	80.38% ± 5.94%	85.58% ± 7.41%	78.46% ± 7.50%
2-d MMDK-means	78.04% ± 8.11%	68.17% ± 11.91%	81.14% ± 6.88%

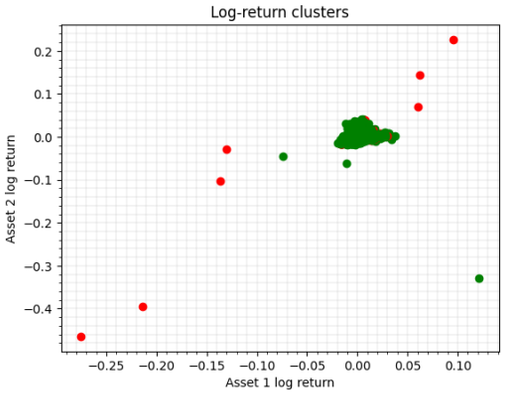
Table 5.15: Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 0.5$, $\rho_1 = 1$, $n = 50$ runs.



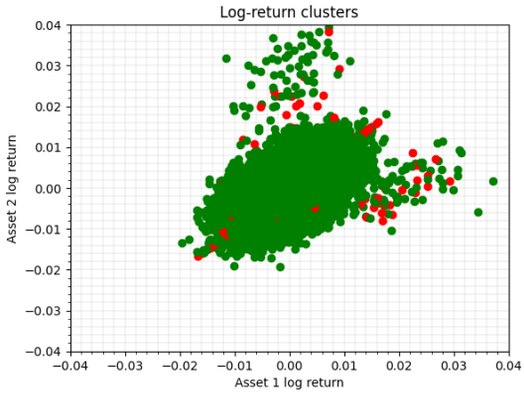
(a) S_1 mean-variance clustering plot.



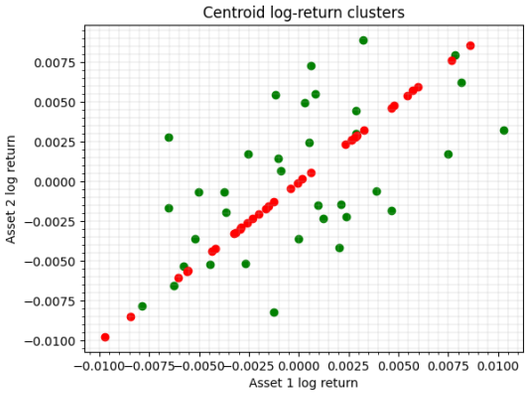
(b) S_2 mean-variance clustering plot.



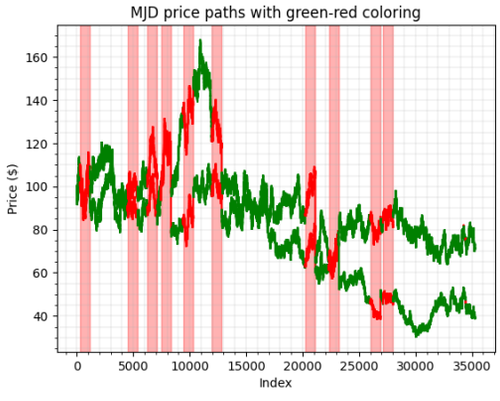
(c) Clusters 1 and 2 log-returns plot with cluster colours.



(d) Clusters 1 and 2 log-returns plot with cluster colours, magnified.



(e) Centroids 1 and 2 log-returns plot with cluster colors.



(f) Price series segmented by cluster association.

Figure 5.16: MJD with correlation regimes, $\rho_0 = 0.5$ and $\rho_1 = 1$, example mean-variance, correlation and price series plots.

5.2.3 Free correlation and mean-variance regime

Finally, we tested the algorithms while varying the mean-variance regime and correlation regime. We tested three joint market regimes: JR_0 = (Bull, Bull, Normal), JR_1 = (Bull, Bull, Abnormal) and JR_2 = (Bear, Bear, Normal). We generated Merton Jump diffusion paths for each asset using the parameter sets

$$\theta_0 = (0.05, 0.05, 0.2, 0.2, 5, 0.02, 0.02, 0.005, 0.005, 0),$$

$$\theta_1 = (0.05, 0.05, 0.2, 0.2, 5, 0.02, 0.02, 0.005, 0.005, 1), \quad \text{and}$$

$$\theta_2 = (-0.05, -0.05, 0.35, 0.35, 10, -0.04, -0.04, 0.04, 0.04, 0).$$

We took $S_0^i = 100$ for $i = 1, 2$, $J = 2$ and $r^1 = r^2 = 5$. We ran the 2-d algorithms for $k = 3$ clusters. We report the accuracy scores for each algorithm in table 5.16 for a total of $n = 50$ runs.

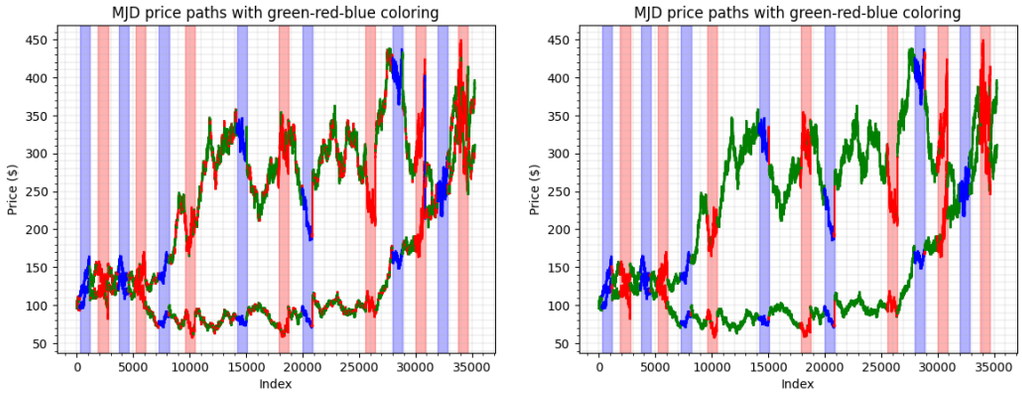
Algorithm	Total	Regime-on	Regime-off
2-d WK-means	78.39% ± 4.47%	64.70% ± 10.05%	85.54% ± 4.86%
2-d MMDK-means	84.31% ± 4.61%	74.20% ± 8.68%	89.52% ± 4.43%

Table 5.16: Accuracy scores with 95% CI, MJD synthetic path, $k = 3$, with correlation and mean-variance regimes, $n = 50$ runs.

Algorithm	Regime-on JR_1	Regime-on JR_2
2-d WK-means	70.68% ± 11.22%	58.71% ± 11.62%
2-d MMDK-means	68.41% ± 12.23%	79.98% ± 8.73%

Table 5.17: Accuracy scores with 95% CI, MJD synthetic path, $k = 3$, with correlation (JR_1) and mean-variance (JR_2) regimes, $n = 50$ runs.

We note the drop in accuracy as seen in table 5.16 when changing both the mean-variance and correlation regime compared to changing only one, which is similar to our observations in the GBM case. Both algorithms struggle to cluster for both regimes and often pick up one whilst ignoring the other as exemplified by figure 5.17(a). We note again the tendency of the 2-d WK-means algorithm to pick up changes in the correlation regime and the 2-d MMDK-means algorithm's tendency to pick up changes in the mean-variance regime as seen in table 5.17. Figure 5.17(b) is an example of a clustering where each type and period of regime change has been picked up successfully. Green segments are those generated by JR_0, red are those generated by JR_1 and blue are those generated by JR_2.



(a) Price series segmented by cluster association, incorrect clustering.

(b) Price series segmented by cluster association, correct clustering.

Figure 5.17: MJD price paths with mean-variance and correlation regimes that do not overlap segmented by cluster association, $\rho_0 = 0$ and $\rho_1 = 1$, example plots.

We then tested four joint regimes: JR_0 = (Bull, Bull, Normal), JR_1 = (Bull, Bull, Abnormal), JR_2 = (Bear, Bear, Normal) and JR_3 = (Bear, Bear, Abnormal). We generated paths for each asset using the parameter sets

$$\theta_0 = (0.05, 0.05, 0.2, 0.2, 5, 0.02, 0.02, 0.005, 0.005, 0),$$

$$\theta_1 = (0.05, 0.05, 0.2, 0.2, 5, 0.02, 0.02, 0.005, 0.005, 1),$$

$$\theta_2 = (-0.05, -0.05, 0.35, 0.35, 10, -0.04, -0.04, 0.04, 0.04, 0), \quad \text{and}$$

$$\theta_3 = (-0.05, -0.05, 0.35, 0.35, 10, -0.04, -0.04, 0.04, 0.04, 1).$$

We took $S_0^i = 100$ for $i = 1, 2$, $J = 3$ and $r^1 = r^2 = r^3 = 5$. We ran the algorithms for $k = 4$ clusters. We report the accuracy scores for each algorithm in table 5.18 for a total of $n = 50$ runs.

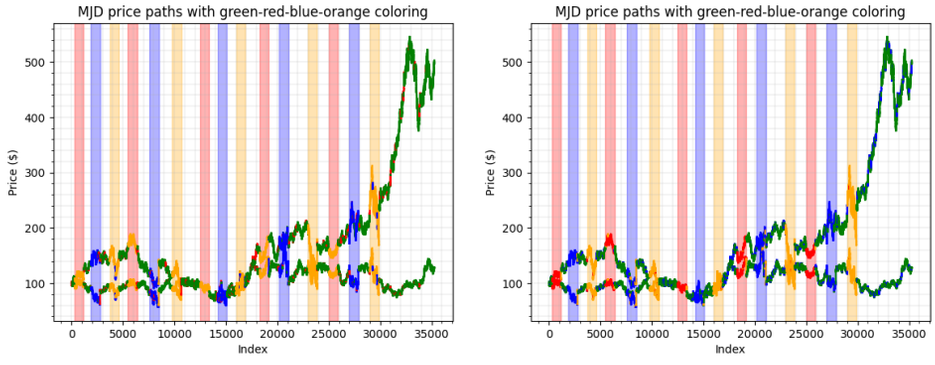
Algorithm	Total	Regime-on	Regime-off
2-d WK-means	68.25% ± 4.34%	53.94% ± 5.37%	76.63% ± 5.68%
2-d MMDK-means	70.76% ± 5.12%	66.78% ± 5.20%	72.95% ± 6.26%

Table 5.18: Accuracy scores with 95% CI, MJD synthetic path, $k = 4$, with correlation and mean-variance regimes, $n = 50$ runs.

Algorithm	Regime-on JR_1	Regime-on JR_2	Regime-on JR_3
2-d WK-means	48.52% ± 13.09%	47.11% ± 10.89%	66.18% ± 12.25%
2-d MMDK-means	31.75% ± 12.38%	76.23% ± 9.89%	92.36% ± 2.83%

Table 5.19: Accuracy scores with 95% CI, MJD synthetic path, $k = 3$, with correlation (JR_1), mean-variance (JR_2) and joint correlation-mean-variance (JR_3) regimes, $n = 50$ runs.

Our accuracy scores in table 5.19 again reflect the trends noted in the GBM case. The algorithms struggle to cluster for both regimes where one of the mean-variance or correlation regime changes (JR_1 and JR_2 respectively) with quite a wide confidence interval and low score. However, both the 2-d WK-means and 2-d MMDK-means algorithms are better at picking up instances of simultaneous change in both the mean-variance and correlation regimes, returning accuracy scores of above 66% and 99% respectively for JR_3. The 2-d MMDK-means algorithm again exceeds in detecting changes in the mean-variance regime as seen in table 5.19 but for both algorithms the overall scores remain low as seen in table 5.18. Figures 5.18(a) and 5.18(b) are examples of our price series coloured according to cluster membership. In figure 5.18(a) the algorithm has successfully identified the presence of certain changes in the market regimes but it struggles to disentangle them and assign them to independent clusters. 5.18(b) is similar but on this occasion the algorithm has managed to identify each of the independent market regime changes.



(a) Price series segmented by cluster association, incorrect clustering.

(b) Price series segmented by cluster association, correct clustering.

Figure 5.18: MJD price paths with mean-variance and correlation regimes segmented by cluster association, $\rho_0 = 0$ and $\rho_1 = 1$, example plots.

5.3 Speed

In this section, we consider the speed of our new algorithms and the impact of increasing the number of market years T, runs n, clusters k, and assets d. We primarily focus on the 2-d WK-means algorithm in our experiments and fix the hyperparameters $(h_1, h_2) = (35, 28)$. In chapter 4 we gave a full description of our 2-d WK-means algorithm and we showed that the 2-Wasserstein distance can be reformulated as a linear sum assignment problem (LSAP) while solving for the 2-Wasserstein barycentre can be done using a distance matrix. Although algorithms solving the LSAP are worst-case $\mathcal{O}(h_1^3)$, use of the *cdist* and *linear_sum_assignment* functions from the *scipy* package give an acceptable speed for our experiments.

The first column of table 5.20 shows the average computation time across 50 runs with $T = 5$ years of hourly data for computing the 2-Wasserstein distance, $\mathcal{W}_{ij}$, and the MMD, MMD_{ij}, between two measures μ_i and μ_j with i, j in $\{1, \ldots, M\}$, for a different number of assets d. Inspecting the computation times for $\mathcal{W}_{ij}$, we might suggest that the computation time tends to remain largely the same as d increases. Indeed, when we inspect the computation time needed for the distance matrix W, shown in the fourth column of table 5.20 and formed using $\mathcal{W}_{ij}$, the total computation time remains relatively unchanged as d increases with a marginal uptick for $d = 100$ assets. In contrast, although the computation time for MMD_{ij} also remains stable irrespective of the number of assets d, it is significantly slower than the $\mathcal{W}_{ij}$ case with computation of $\mathcal{W}_{ij}$ approximately an order of magnitude faster. When conducting experiments, this difference in speed impacted computation of the distance matrix, with calculations involving $\mathcal{W}_{ij}$ being significantly quicker than those involving MMD_{ij} in our test environments.

d	$\mathcal{W}_{ij}$	MMD_{ij}	Distance matrix (W)
2	2.05s $\times 10^{-5}$ ± 3.41s $\times 10^{-6}$	4.70s $\times 10^{-4}$ ± 9.78s $\times 10^{-5}$	27.91s ± 0.05s
3	2.91s $\times 10^{-5}$ ± 1.81s $\times 10^{-5}$	3.79s $\times 10^{-4}$ ± 6.77s $\times 10^{-5}$	27.35s ± 0.03s
5	7.17s $\times 10^{-5}$ ± 5.85s $\times 10^{-5}$	4.50s $\times 10^{-4}$ ± 6.04s $\times 10^{-5}$	27.00s ± 0.06s
10	4.78s $\times 10^{-5}$ ± 1.79s $\times 10^{-5}$	5.29s $\times 10^{-4}$ ± 1.28s $\times 10^{-4}$	27.42s ± 0.04s
20	4.06s $\times 10^{-5}$ ± 1.21s $\times 10^{-5}$	4.53s $\times 10^{-4}$ ± 7.45s $\times 10^{-5}$	27.33s ± 0.02s
100	3.24s $\times 10^{-5}$ ± 1.21s $\times 10^{-5}$	4.28s $\times 10^{-4}$ ± 1.30s $\times 10^{-4}$	29.05s ± 0.03s

Table 5.20: Distance matrix computation time with 95% CI for d assets, $T = 5$.

The real bottleneck of our algorithm is the distance matrix W, computed to solve for the 2-Wasserstein and MMD barycentre. Calculating the distance matrix W is $\mathcal{O}(\frac{M(M-1)}{2})$ where M is the number of measures formed from our segmented stream of returns of length $(T \times 252 \times 7) - 1$. The computation time for computing W for different values of T shown in table 5.21 reflects this, and we see that a single run can take approximately 7.5 minutes for $T = 20$ market years of hourly data. Table 5.22 demonstrates the average time taken to complete n runs for each value of T, *post*-calculation of the distance matrix W, calculated as an average of 50 such computations. We note that increasing the number of runs does not exceptionally increase the duration of the algorithm. This is one advantage of using the distance matrix when computing the barycentre. We need only compute W once and the subsequent time to complete n runs is approximately $\mathcal{O}(n)$.

T	Row i $([\mathcal{W}_{ij}]_{1 \leq j \leq M})$	Distance matrix (W)
2	0.019s ± 0.0002s	4.47s ± 0.01s
5	0.048s ± 0.0003s	27.91s ± 0.05s
10	0.096s ± 0.0003s	113.36s ± 0.35s
20	0.197s ± 0.0005s	454.33s ± 1.10s

Table 5.21: Computation time with 95% CI for computing one row of the distance matrix $([\mathcal{W}_{ij}]_{1 \leq j \leq M})$, and for the entire distance matrix W, for T years of data and over 50 runs.

T \ n	1	5	25	50
2	0.31s ± 0.02s	1.48s ± 0.06s	7.12 ± 0.11	14.32s ± 0.15s
5	0.68s ± 0.06s	3.60s ± 0.16s	18.25s ± 0.32s	36.02s ± 0.48s
10	1.55s ± 0.13s	7.63s ± 0.30s	37.28s ± 0.83s	74.28s ± 1.04s
20	3.05s ± 0.25s	15.47s ± 0.57s	78.72s ± 1.32s	157.22s ± 1.95

Table 5.22: Computation time with 95% CI for *post*-distance matrix computation for n runs of the 2-d WK-means algorithm, for T years of data, and where $k = 2$ computed over 50 runs.

If we were to use this algorithm in a production environment, we would ideally like the computation time to be shorter. Another advantage of using the distance matrix is that we do not need to recreate the matrix each time we run our algorithm. Instead, we may store each row, as shown in figure 4.1, for future use. Upon receipt of a new empirical measure, which we call μ_{M+1}, we proceed to calculate its distance to each of the previous measures, $([\mathcal{W}_{(M+1)j}]_{1\leq j\leq M})$ and append these distances to each respective row. We also append the new row $([\mathcal{W}_{(M+1)j}]_{1\leq j\leq M})$. Computing the total distance between any one measure and all prior measures is significantly faster and approximately $\mathcal{O}(M)$, as shown in column two of table 5.21.

Finally, table 5.23 shows the computation time for running the iteration *post*-distance matrix computation for different numbers of clusters k and runs n. In order to assign each measure to its respective cluster our algorithm must compute the distance between each measure and each cluster centroid. Increasing the number of clusters will then lead to more such computations and a longer run time, which is borne out in the results in table 5.23.

k \ n	1	5	25
2	2.59s ± 0.11s	13.22s ± 0.34s	65.88s ± 0.88s
4	3.95s ± 0.23s	20.34s ± 0.57s	97.61s ± 1.25s
8	7.38s ± 0.45s	36.87s ± 0.94s	183.89s ± 2.19s

Table 5.23: Total computation time with 95% CI for k clusters and n runs, $T = 20$, $d = 2$.

5.4 Summary

The following is a short summary of our key findings from our synthetic data experiments which should help inform us of our approach in the real data environment. Both the 2-d WK-means and 2-d MMDK-means algorithms returned strong scores when testing for a change in one of the mean-variance regime or the correlation regime. We found that often the 2-d WK-means algorithm was stronger when detecting different changes in the correlation regime for both the GBM and MJD cases. In contrast, the 2-d MMDK-means algorithm thrived when asked to detect changes in the mean-variance regime, particularly when working with MJD data. The accuracy of both algorithms however drops as we increase the number of different types of regime, particularly when we introduce both changes in the mean-variance and correlation regimes to our data. Finally, we note that computation of the distance MMD_{ij} tends to be slower than that of $\mathcal{W}_{ij}$ for each pair of measures μ_i and μ_j with i, j in $\{1, \ldots, M\}$, thus leading to a longer computation time for the distance matrix W when using the MMD.

Chapter 6

Real data experiments

In this section we consider applications of the 2-d algorithms to sets of real data taken from securities in the S&P 500. We learned in our synthetic data experiments that the 2-d WK-means and 2-d MMDK-means algorithms are effective when clustering mean-variance regimes in the marginal distributions or when clustering correlation regimes between the assets. In such experiments, our algorithms return high regime-on and regime-off accuracy scores. However, the accuracy scores fall significantly when we attempt to cluster both mean-variance and correlation regimes in the same dataset as exhibited in tables 5.9 and 5.18. Thus should we apply the 2-d WK-means or 2-d MMDK-means algorithm to our real data directly, we likely will find the clustering to be weaker than we might otherwise desire.

Therefore we propose a different approach. Instead of applying one of the 2-d algorithms alone, we make use of both the uni-d 1-WK-means algorithm and one of the 2-d algorithms in a best of both worlds approach. We begin by applying the uni-d 1-WK-means algorithm to the data in order to remove the effects of the marginal distribution on each asset and subsequently apply the 2-d WK-means or 2-d MMDK-means algorithm to this transformed data. This method is based on the theory of copulas.

Definition 6.0.1 (Copula, [76] (Definition 1, page 229))**.** Let $X = (X_1, \dots, X_d)$ be a random vector with joint CDF F and continuous marginal CDFs $F_1, \dots, F_d$. Then we say that the copula of F (or X) is the joint CDF C of the random vector $(F_1(X_1), \dots, F_d(X_d))$ on $[0,1]^d$.

The copula of a random vector X captures the dependency structure between the marginals $X_1, \dots, X_d$ and thanks to the work of Sklar we have theorem 6.0.2.

Theorem 6.0.2 (Sklar's theorem, [76] (Theorem 2, page 230))**.** *Let $X = (X_1, \dots, X_d)$ be a random vector with joint CDF F and with marginal CDFs $F_1, \dots, F_d$. Then there is a copula $C : [0,1]^d \to [0,1]$ such that*

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

where $x_1, \dots, x_d \in \mathbb{R} \cup \{-\infty, \infty\}$. Moreover, the copula is uniquely defined when $F_1, \dots, F_d$ are continuous.

In our synthetic experiments we characterised changes in the joint regime between two assets by their respective marginal distribution's changes in mean and variance, and their joint change in correlation. Correlation can tell us how closely related our two assets are in a linear sense but the copula structure is far richer, and it helps us model the inter-correlation dependence between assets. If we apply the marginal distribution to each respective asset then the joint distribution of our transformed data should be the copula according to Sklar's theorem. Moreover, the transformed data for each asset should be uniform.

Proposition 6.0.3 (Probability transformation)**.** *Suppose X is a random variable with continuous CDF F. Then $F(X) \sim$ Uniform(0, 1).*

Proof. See appendix A.4.1. □

We now have a set of tools with which to approach our real data clustering problem. Our approach when treating real data is laid out in the following steps.

Step 1: Univariate clustering and data transformation

We begin by establishing the marginal distributions of each asset using the uni-d WK-means algorithm. As established in our synthetic data experiments, tables 5.2 and 5.11, as well as [1], the uni-d WK-means algorithm performs well in detecting the underlying probability distributions, characterised by their mean and variance, present in the data of an individual asset. In particular, we make use of the uni-d 1-WK-means algorithm due to its superior accuracy when testing on data generated from a Merton jump diffusion process. In order to estimate the number of clusters present in our uni-dimensional data, we propose an approach based on the Kolmogorov-Smirnov test as described in theorem 6.0.4.

Theorem 6.0.4 (Kolmogorov-Smirnov two-tail goodness-of-fit test, [77] (pages 283-287))**.** *Let $X = (X_1, \ldots, X_n)$ be independent and identically distributed random variables with an empirical cumulative distribution function denoted $F_n(x)$ and let $F(x)$ be a continuous distribution. We set up a goodness-of-fit test using the Kolmogorov-Smirnov test statistic defined as*

$$D_n = sup_x |F_n(x) - F(x)|,$$

and where our hypotheses to test are

$$\begin{aligned} H_0 &: F_n(x) = F(x) \quad \textit{for each } x, \\ H_1 &: F_n(x) \neq F(x) \quad \textit{for at least one value of } x. \end{aligned}$$

We first assume that each asset has the minimum number of clusters possible, $k = 2$. If we have chosen the correct number of clusters for the given data then we would expect the uni-dimensional clustering to be effective. Furthermore, if we were to then apply the empirical cumulative distribution function of each cluster to its respective returns data, proposition 6.0.3 assures us that the transformed returns data should in fact be uniform. Therefore application of the Kolmogorov-Smirnov test, theorem 6.0.4 where $F(x)$ is a uniform distribution, should lead us to not reject the null hypothesis that our data is uniform. Since we are using the empirical CDF formed from all values in each cluster, we would expect the p-value yielded from our test to be close to 1. Should we find the p-value to not be close to 1, we might reason that a larger number of clusters may be required. Subsequently, we increase the number of clusters k until the p-value is very close to or exactly 1.

Having established approximations to the marginal distributions for each asset, we then proceed to apply the empirical cumulative distribution function of our approximations to the data in their respective clusters. As previously discussed, the process of segmenting our data for clustering involves overlapping data and thus any one return may belong to up to 5 different segments. Each of these segments is then placed in one of our clusters and so, any one return can be associated with possibly multiple centroids and we then have a choice of which empirical CDF to apply. We follow our previous approach in this regard, and associate each return with the first cluster to which it is assigned. We now have a new dataset of transformed returns where each value falls within [0, 1] and should follow a uniform distribution.

Step 2: Multivariate clustering

The joint distribution of our newly transformed dataset should be the copula of the original data, and so we apply the 2-d WK-means or 2-d MMDK-means algorithm to our transformed data in order to obtain correlation regimes.

6.1 Synthetic data experiments revisited

In order to validate our approach, we will briefly return to our previous synthetic data experiments, specifically to those of the free correlation and mean-variance regime clustering problems introduced in sections 5.1.3 and 5.2.3. Experimenting with both GBM and MJD price paths, we found that the accuracy scores suffered when either of the 2-d algorithms was tasked with clustering mean-variance regime changes in the marginal distributions as well as correlation regime changes across both assets using four clusters, as shown in tables 5.10 and 5.19. We treat both experiments using our new two-step approach, beginning with the case of the GBM price path in some detail and subsequently for the MJD.

Step 1: Univariate clustering

We first cluster each of our two assets using the uni-d 1-WK-means algorithm where we take $k = 2$. As in our synthetic data experiment, we display mean-variance clustering plots in figures 6.1(a) and 6.1(b). Upon inspection, we see that the algorithm appears to have performed well in clustering for both regimes. The higher variance cluster clearly has a lower mean than that of the lower variance cluster for both assets, as seen by the placement of the cluster cross on the plots. Table 6.1 displays the accuracy scores achieved for each asset, confirming that this is indeed a strong clustering. Finally, figures 6.1(c) and 6.1(d) display histograms of $\mathbb{E}(\mu_i)_{1\leq i\leq M}$ for the distributions generated from segments of assets 1 and 2, and analogously figures 6.1(e) and 6.1(f) show histograms of $\mathbb{V}(\mu_i)_{1\leq i\leq M}$. The histogram plots for each asset also show dashed vertical lines indicating the value of each centroid's mean and variance, and a solid line indicating the value of each regime's theoretical mean and variance. Recalling the mean and variance of the parameter sets

$$\theta_0 = (0.02, 0.02, 0.2, 0.2), \quad \text{and} \quad \theta_1 = (-0.02, -0.02, 0.3, 0.3)$$

used to simulate the GBM and generate each asset's log-returns for the joint regimes JR_0 (bull) and JR_1 (bear), the true distributions of the log-returns are

$$\mathbb{P}_{\text{JR}_0} = \mathcal{N}(-1.97 \times 10^{-21}, 2.27 \times 10^{-05}), \quad \text{and} \quad \mathbb{P}_{\text{JR}_1} = \mathcal{N}(-3.68 \times 10^{-05}, 5.1 \times 10^{-05})$$

for each asset, where $\mathcal{N}$ is the normal distribution. It appears that the first two moments of our centroids match the theoretical values quite well and thus, the empirical CDF of each cluster should act as a sufficient proxy for the true distributions in step 2.

Algorithm	Asset	Total	Regime-on	Regime-off
Uni-d 1-WK-means	1	94.49%	94.43%	94.29%
Uni-d 1-WK-means	2	95.82%	96.06%	95.52%

Table 6.1: Accuracy scores of uni-d 1-WK-means applied to GBM synthetic paths with mean-variance and correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, $n = 50$ runs.

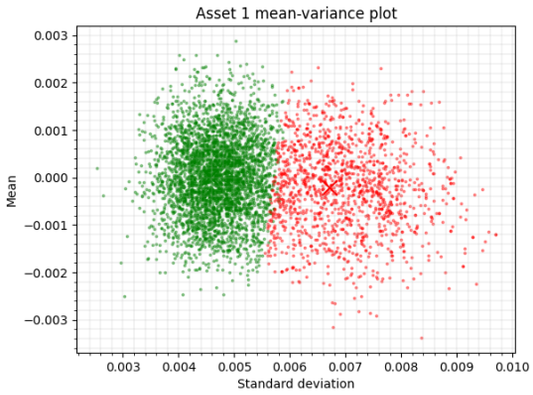
We now apply the empirical cumulative distribution function of each cluster to their respective cluster of log-returns. Conducting the Kolmogorov-Smirnov test on each asset's transformed returns yields p-values of 1.000 and 1.000 respectively. In this instance we have the advantage of foresight and hence we know that $k = 2$ is the correct number of clusters. In our real data experiments, this step often requires changing the number of clusters k until we have a satisfactory p-value.

Step 2: Multivariate clustering

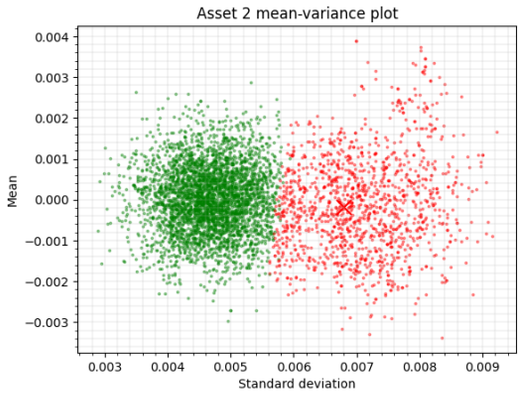
Finally, we apply the 2-d WK-means and 2-d MMDK-means algorithms to the transformed data and display the accuracy scores in table 6.2

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	99.48% ± 0.00%	99.01% ± 0.00%	99.41% ± 0.00%
2-d MMDK-means	99.44% ± 0.00%	99.41% ± 0.00%	99.22% ± 0.00%

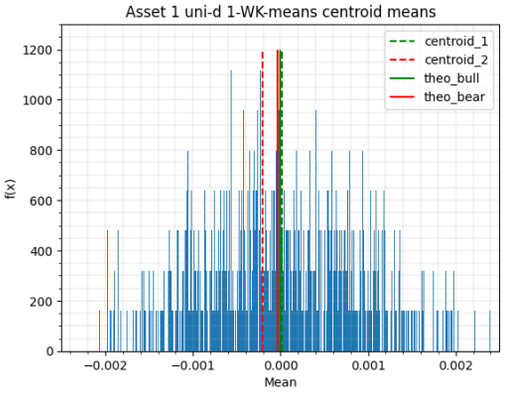
Table 6.2: Accuracy scores of 2-d WK-means and 2-d MMDK-means applied to GBM synthetic paths with mean-variance and correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, $n = 50$ runs.



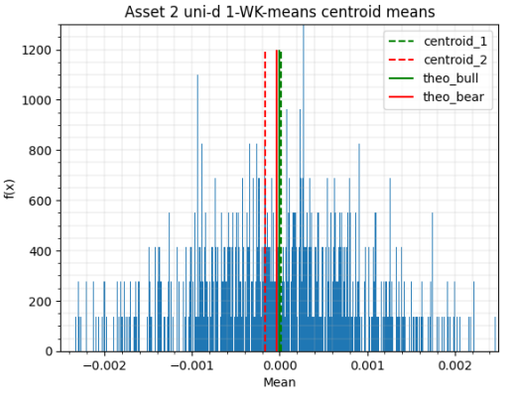
(a) S_1 mean-variance clustering plot.



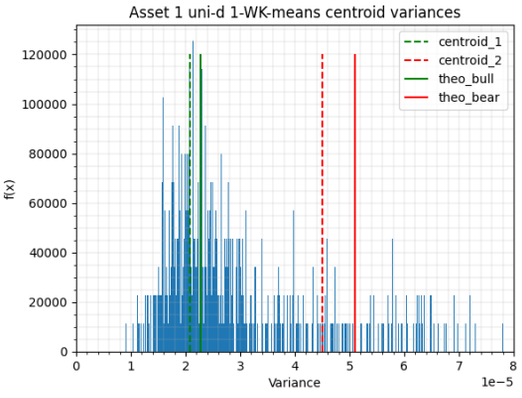
(b) S_2 mean-variance clustering plot.



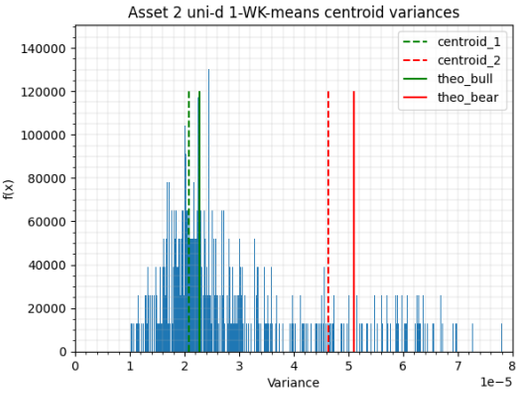
(c) S_1 distribution of $\{\mathbb{E}(\mu_i)\}_{i\geq 0}$.



(d) S_2 distribution of $\{\mathbb{E}(\mu_i)\}_{i\geq 0}$.



(e) S_1 distribution of $\{\mathbb{V}(\mu_i)\}_{i\geq 0}$.



(f) S_2 distribution of $\{\mathbb{V}(\mu_i)\}_{i\geq 0}$.

Figure 6.1: Uni-d 1-WK-means applied to assets 1 and 2 clustering and centroid plots.

Clearly, when compared to our original results in table 5.10, we have a substantial improvement in our regime accuracy scores. Figure 6.2 displays the associated price series colored according to cluster association. Figures 6.2(c) and 6.2(d) show the colored price series for assets 1 and 2 respectively with periods of mean-variance regime change highlighted in red. Similarly, figure 6.2(e) shows both segmented time series with periods of correlation regime change highlighted.

We also introduce two new plots in figures 6.2(a) and 6.2(b) that we may use to analyse our results. Figure 6.2(a) shows the correlation between the two assets over a rolling window of 35 market hours. Each point is then assigned a color based on the cluster association of the returns it is formed from. It is noticeable that those periods of high rolling correlation are most commonly associated with cluster 2, which is colored red, as we would expect. All other points are colored green indicating that their associated returns belong to cluster 1.

Figure 6.2(b) shows the correlation of each of the empirical measures formed from our returns data, colored according to their cluster association. We note that periods of increasing correlation in the empirical measures are colored red while periods of decreasing or relatively stationary correlation are colored green. This is in line with what we might expect and further supports our claim that the 2-d WK-means algorithm has effectively clustered the empirical measures into their correct correlation regimes. These plots are reassuring when viewed in the context of our synthetic examples but they will become much more useful when we identify the correlation regimes present in real data which we do not know *a priori*.

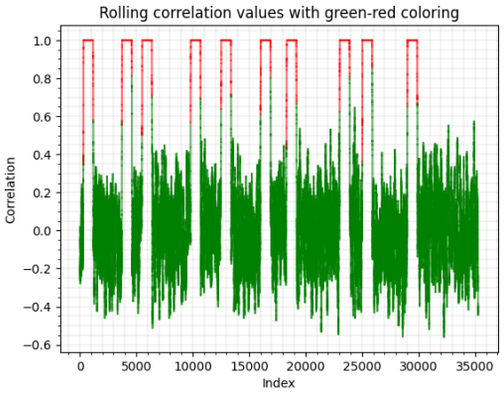
Analogously, we apply our approach to the price series generated by the Merton jump diffusion process in section 5.2.3. The accuracy scores are summarised in tables 6.3 and 6.4. Figure 6.3 shows the associated color plots.

Algorithm	Asset	Total	Regime-on	Regime-off
Uni-d 1-WK-means	1	97.71%	97.30%	97.62%
Uni-d 1-WK-means	2	97.14%	97.59%	96.77%

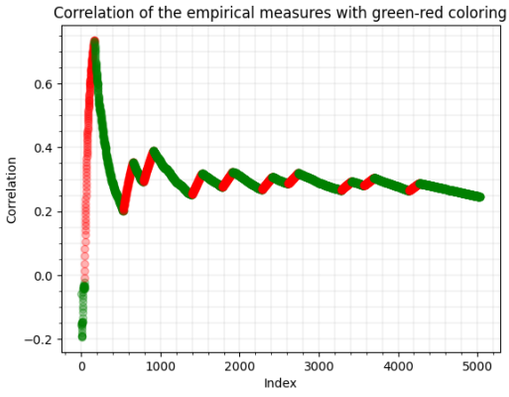
Table 6.3: Accuracy scores of uni-d 1-WK-means applied to MJD synthetic paths with mean-variance and correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	99.46% $\pm$ 0.00%	98.69% $\pm$ 0.00%	99.49% $\pm$ 0.00%
2-d MMDK-means	99.42% $\pm$ 0.00%	99.44% $\pm$ 0.00%	99.18% $\pm$ 0.00%

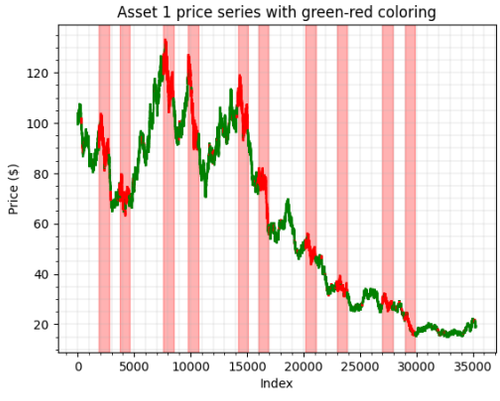
Table 6.4: Accuracy scores of 2-d WK-means and 2-d MMDK-means applied to MJD synthetic paths with mean-variance and correlation regimes, $\rho_0 = 0$ and $\rho_1 = 1$, $n = 50$ runs.



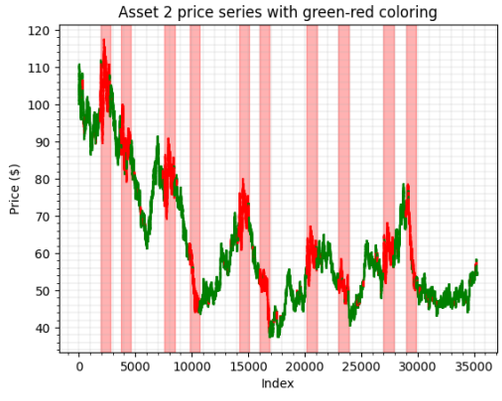
(a) Correlation between assets 1 and 2 over a rolling window, coloured according to cluster designation.



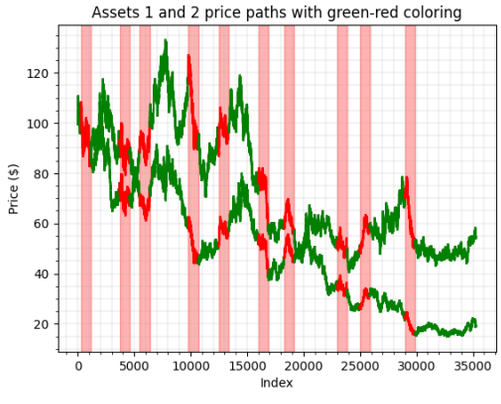
(b) Correlation of each empirical measure, coloured according to cluster designation.



(c) S_1 price series segmented by mean-variance regime cluster association.

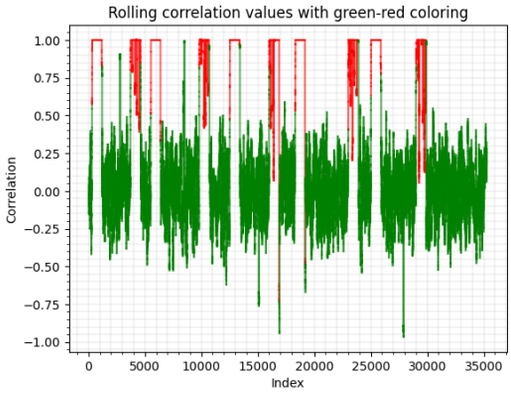


(d) S_2 price series segmented by mean-variance regime cluster association.

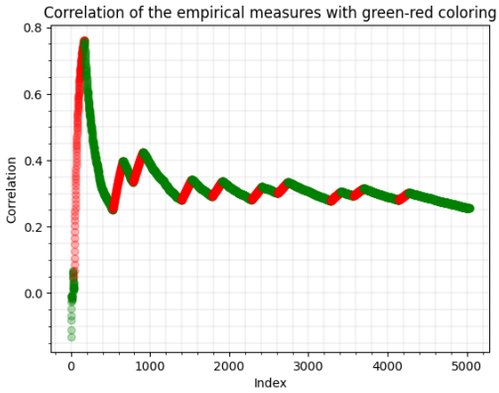


(e) S_1 and S_2 price series segmented by correlation regime cluster association.

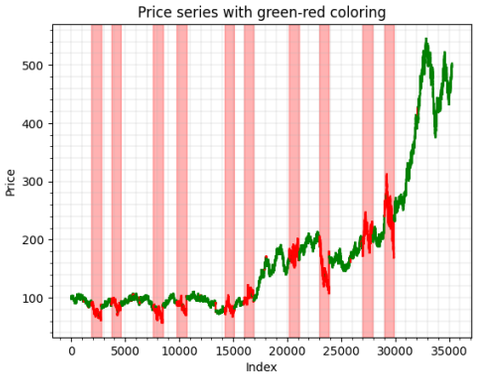
Figure 6.2: Geometric Brownian motion assets 1 and 2 correlation and price series segmented by cluster association.



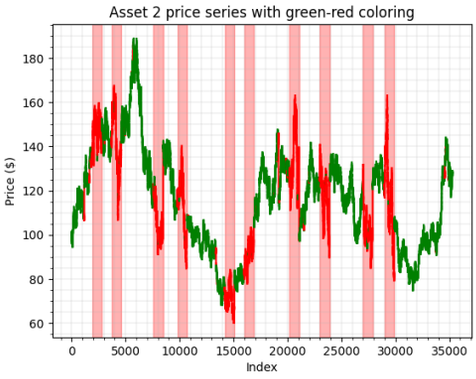
(a) Correlation between assets 1 and 2 over a rolling window, coloured according to cluster designation.



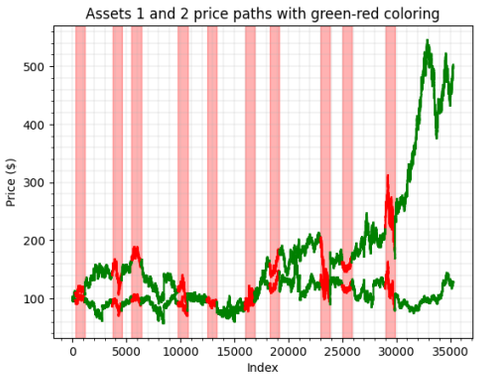
(b) Correlation of each empirical measure, coloured according to cluster designation.



(c) S_1 price series segmented by mean-variance regime cluster association.



(d) S_2 price series segmented by mean-variance regime cluster association.



(e) S_1 and S_2 price series segmented by correlation regime cluster association.

Figure 6.3: Merton jump diffusion assets 1 and 2 correlation and price series segmented by cluster association.

6.2 Real data

Now that we have validated our approach on our synthetic examples, we proceed to treat real world cases. At this point we diverge slightly from our previous experiments and use only the 2-d WK-means algorithm going forward. The reason for doing so is three-fold:

1. The primary function of the 2-d algorithm in our two-step approach is to identify correlation regimes in the data. In our synthetic data experiments we found the 2-d WK-means algorithm to often be superior in this domain.

2. Although not unreasonable, the speed of computation for the distance matrix W is significantly slower when using the Maximum Mean Discrepancy.

3. Use of the MMD requires choosing a bandwidth parameter, σ. Thus far we have averaged our results over bandwidth parameters in the range of 10^{-3} to 10^{-6}. It is possible that further research may yield an optimal σ which may yet prove more effective when clustering but for the moment the estimation remains basic. The 2-d WK-means algorithm requires no such parameter.

Our subsequent experiments will be conducted using three pairs of stocks; Apple Inc and Amazon.com Inc (AAPL-AMZN), AvalonBay Communities Inc and Essex Property Trust Inc (AVB-ESS), and J.M. Smucker Co and Paycom Software Inc (SJM-PAYC). In most cases of pairs trading, stocks are selected such that they have some innate relationship of which the trader is typically aware. For example, AVB-ESS would be a possible candidate pairing as both stocks represent real estate companies with typically high correlation, and the trader may use this property, or some other relationship such as possible cointegration, in order to create a profitable strategy.

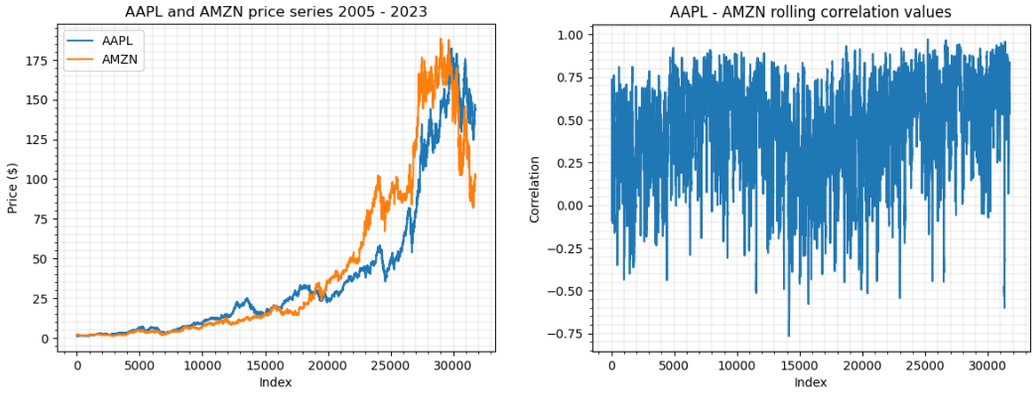
This is not our approach. We do not rely on any preconceived notions of stocks in the S&P 500 to form our portfolios. So far as we have tested our ideas, the general framework laid out at the start of this chapter works in principle for any pairing. The three pairings chosen are used to demonstrate the ubiquity of this approach irrespective of the general correlation between the two assets. AVB-ESS are highly correlated, while SJM-PAYC have a correlation close to 0 and AAPL-AMZN have a correlation just below 0.5 across their respective testing periods. We will go through the steps in some detail when approaching the case of AAPL-AMZN, before providing shorter summaries for the other pairings. We will naively use hyperparameters $(h_1, h_2) = (35, 28)$ for both the uni-d 1-WK-means and 2-d WK-means algorithms before providing a more detailed investigation in section 6.3.

6.2.1 AAPL - AMZN

No modern finance-related paper would be complete without some mention of a tech stock. Luckily, we get to mention two: Apple Inc (AAPL) and Amazon.com Inc (AMZN). We use the hourly close prices for each stock from 2005-2023. The correlation of these assets over the entire testing period is 0.433 while the standard deviation of their rolling correlation using a window of 35 hours is 28.68%. Figure 6.4(a) shows the price series evolution of AAPL and AMZN stock between 2005 and 2023 while figure 6.4(b) shows the rolling correlation over a window of 35 market hours between these two assets during that time.

Step 1: Univariate clustering and data transformation

We initially cluster the empirical measures of each of our two assets using the uni-d 1-WK-means algorithm. We begin by choosing the smallest number of clusters possible, $k = 2$ clusters, for each asset, such that each cluster represents either a low or high variance regime. Subsequently, we then associate each return to the cluster that its first empirical measure is placed in. We then gather the returns into two groups representing the two clusters, and apply the empirical CDF of each cluster to its constituent data, thus leaving us with transformed data where each value is between 0 and 1. In this case, since we have chosen $k = 2$, if there are two underlying probability distributions in the returns of each asset and our clustering has captured these, then we would expect the transformed data to be uniform.



(a) AAPL and AMZN price series 2005 - 2023.

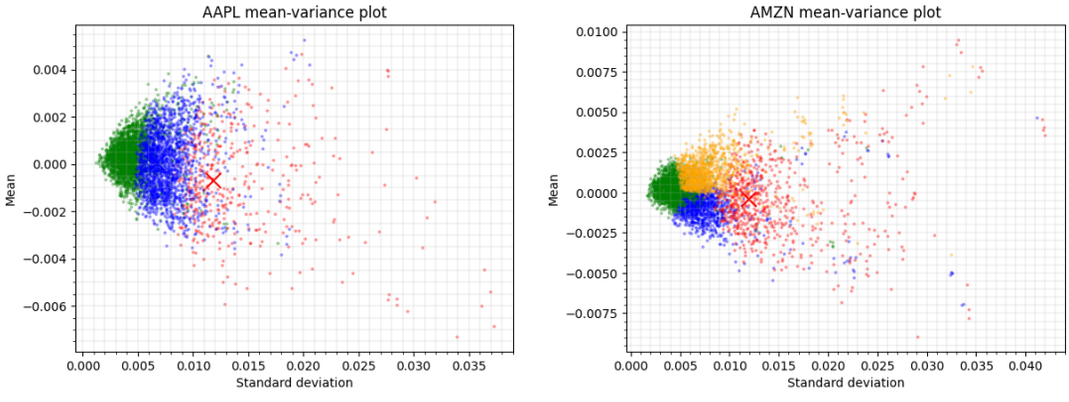
(b) AAPL and AMZN rolling correlation of returns plot using a rolling window of 35 market hours.

Figure 6.4: Price series and rolling correlation plots of AAPL and AMZN.

To test the uniformity of our transformed data, we apply the Kolmogorov-Smirnov test. Given we are using the empirical CDF of all the data in each cluster, we would expect the p-value of our test to be close to 1. If it is less than 1, it is possible that we require a further cluster. This methodology is critical to determining the number of clusters required for transforming our data. If we do not remove the marginal distributions from our data then the 2-d WK-means algorithm employed in our second step will struggle to detect the copula structure and changes in correlation, as seen in our synthetic experiments 5.1.3 and 5.2.3.

In the case of AAPL and AMZN, should we use $k = 2$ clusters our tests return a p-value of 0.944 and 0.801 respectively. We therefore increase the number of clusters, and find that $k = 3$ clusters returns a p-value of 0.999 for AAPL while $k = 4$ clusters returns a p-value of 0.982 for AMZN.

Figures 6.5(a) and 6.5(b) show the mean-variance clustering plots of each asset. AAPL seems to exhibit three mean-variance regimes of increasing variance; a low variance regime (green), a medium variance regime (blue) and a high variance regime (red). AMZN also exhibits three regimes of increasing variance, however its medium variance regime is further split into two regimes of high (orange) and low (blue) means.



(a) AAPL mean-variance clustering plot.

(b) AMZN mean-variance clustering plot.

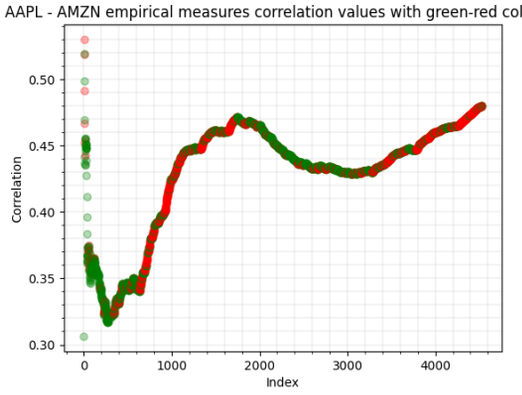
Figure 6.5: Uni-d 1-WK-means applied to AAPL and AMZN mean-variance clustering plots.

Step 2: Multivariate clustering

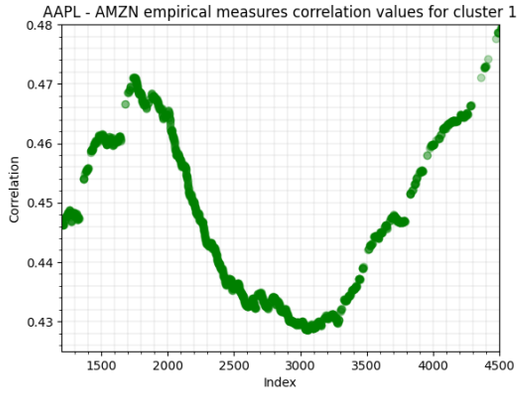
Now that we have removed the influence of the marginal distributions on the data, we apply the 2-d WK-means algorithm. As an input to our algorithm, we must decide on the number of clusters k we will run our algorithm for. The simplest possible choice is $k = 2$. These would represent one regime of higher correlation and one of a lower correlation.

Indeed, for $k = 2$, we obtain one correlation regime of high correlation where the correlation of the centroid's atoms is 0.738 and one of a lower correlation, where the correlation of the centroid's atoms is 0.373. Figure 6.6(a) shows the correlation of each of the empirical measures that we have clustered with their associated cluster colors. Much like our synthetic example plot in figure 6.2(b), we note that where the correlation is increasing the second (red) cluster tends to dominate while where the correlation is decreasing, the first (green) cluster is more apparent. These trends are further reinforced in figures 6.6(b) and 6.6(c) which show a portion of figure 6.6(a) magnified and where only one cluster is presented.

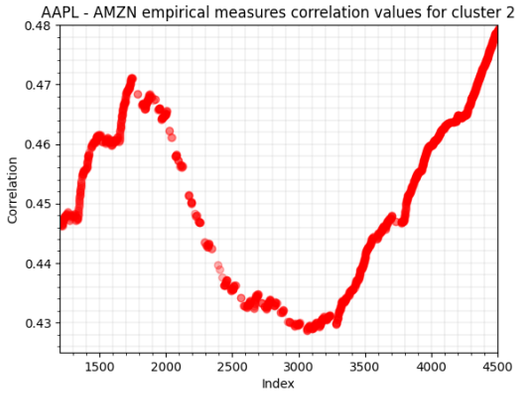
Figures 6.7(a) and 6.7(b) show our standard plots of the correlation of the centroids and log returns respectively of each cluster. Figure 6.7(c) shows the rolling correlation plot with cluster colors. We see from the plot that the second (red) cluster has caught instances of higher correlation above the average line, represented by the black horizontal line, while the first (green) cluster tends to fall below this line indicating that it captures periods of lower correlation. This is reassuring as it is similar to the analogous plot in our synthetic example, figure 6.2(a). Figures 6.7(d) and 6.7(e) show the same plot where only one cluster is shown.



(a) AAPL and AMZN empirical measures correlation plot with cluster colors.

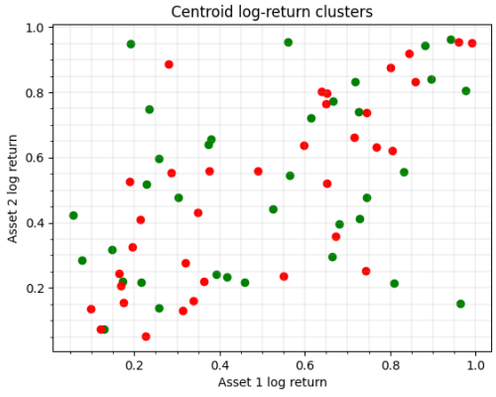


(b) AAPL and AMZN empirical measures correlation plot with cluster colors, cluster 1 only.

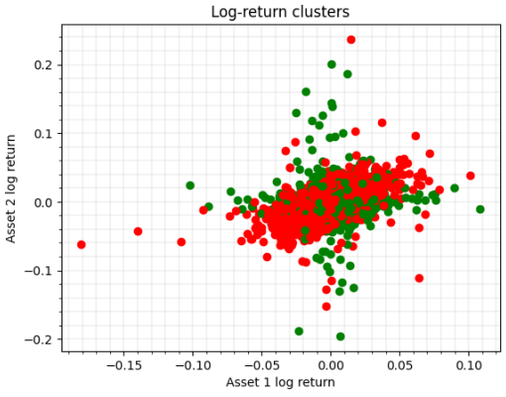


(c) AAPL and AMZN empirical measures correlation plot with cluster colors, cluster 2 only.

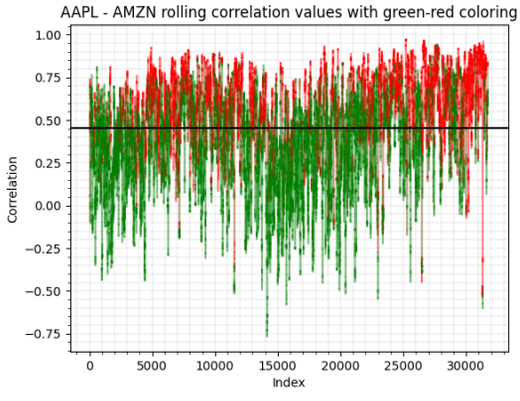
Figure 6.6: Empirical measures correlation plots of AAPL and AMZN for 2 clusters.



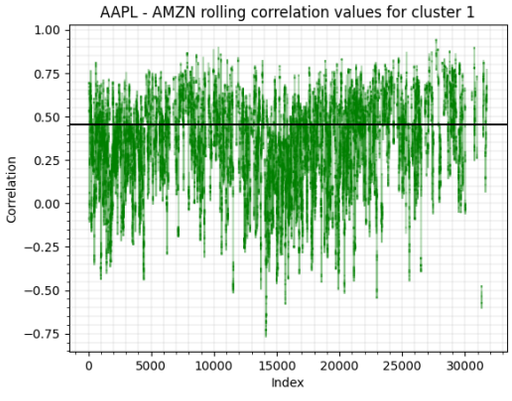
(a) Centroids 1 and 2 log-returns plot with cluster colors.



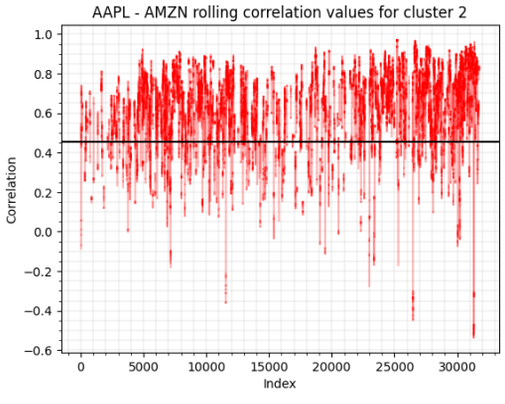
(b) Clusters 1 and 2 log-returns plot with cluster colors.



(c) AAPL and AMZN rolling correlation of returns plot using a rolling window of 35 market hours with cluster colors.



(d) AAPL and AMZN rolling correlation of returns plot using a rolling window of 35 market hours with cluster colors, cluster 1 only.

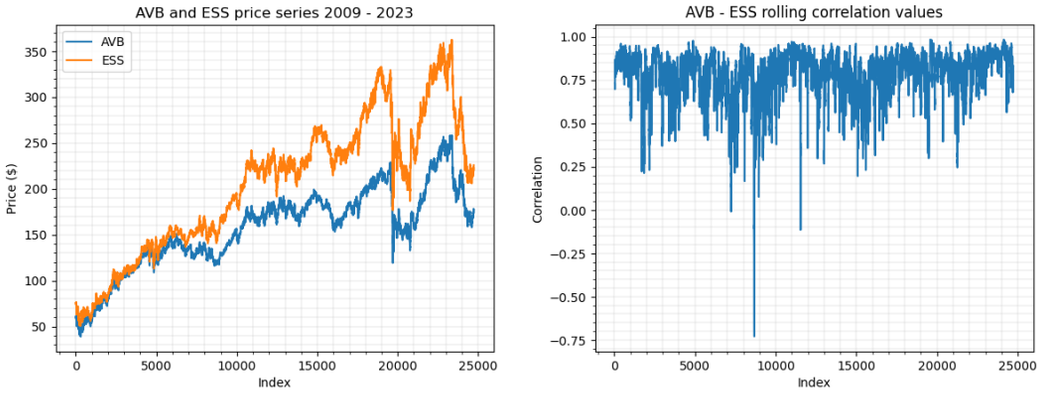


(e) AAPL and AMZN rolling correlation of returns plot using a rolling window of 35 market hours with cluster colors, cluster 2 only.

Figure 6.7: Centroid, cluster and rolling correlation plots of AAPL and AMZN for 2 clusters.

6.2.2 AVB - ESS

AVB and ESS constitute our second test pairing. These stocks represent AvalonBay Communities Inc (AVB) and Essex Property Trust Inc (ESS), both of which are real estate companies and thus we would expect them to be highly correlated. Indeed, using the hourly close prices for each stock from 2009-2023, the correlation of these assets over the entire testing period is 0.835 while the standard deviation of their rolling correlation using a window of 35 market hours is 13.96%. This is a noticeably lower variation in the rolling correlation than that of AAPL-AMZN. Figure 6.8(a) shows the price series evolution of AVB and ESS stock between 2009 and 2023 while figure 6.8(b) shows the rolling correlation over a window of 35 market hours between these two assets during that time.



(a) AVB and ESS price series 2009 - 2023.

(b) AVB and ESS rolling correlation of returns plot using a rolling window of 35 market hours.

Figure 6.8: Price series and rolling correlation plots of AVB and ESS.

Step 1: Univariate clustering and data transformation

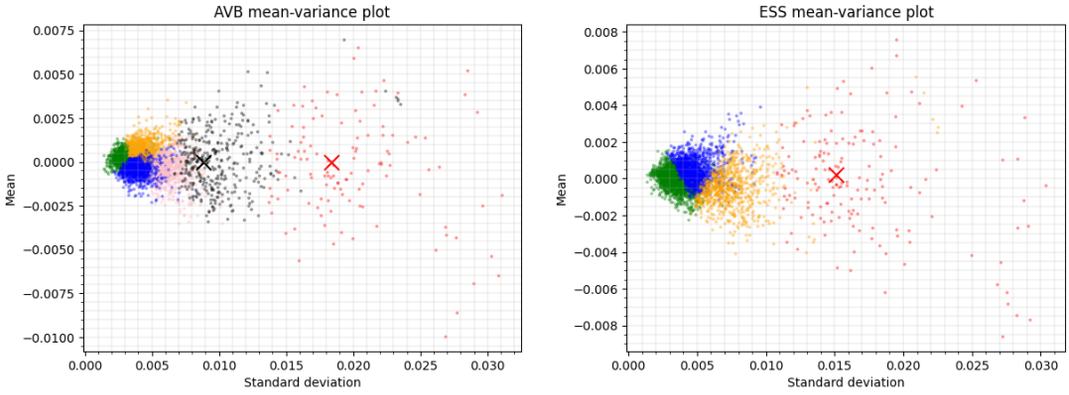
We initially cluster the empirical measures of each of the two assets using the uni-d 1-WK-means algorithm. As with AAPL-AMZN, we begin by choosing the smallest number of clusters possible, $k = 2$ clusters, for each asset. After testing, we find that in the case of AVB and ESS, we require 6 and 4 clusters respectively. When using $k = 6$ for AVB, the Kolmogorv-Smirnov test yields a p-value of 0.994 and when using $k = 4$ clusters for ESS, we have a p-value of 0.991.

Figures 6.9(a) and 6.9(b) show the mean-variance clustering plots of each asset. AVB exhibits six mean-variance regimes; a very low variance regime (green), a low variance regime (blue) with low mean, a low variance regime with high mean (orange), a medium variance regime (pink), a high variance regime (black) and a very high variance regime (red). ESS however exhibits four mean-variance regimes of increasing variance; a low variance regime (green), a medium variance regime (blue), a high variance regime (orange), and a very high variance regime (red).

Step 2: Multivariate clustering

Upon inspection of the rolling correlation graph in 6.8(b), although there are notable downward spikes in correlation at times, the rolling correlation tends to be quite steady on average. It therefore seems reasonable to take $k = 2$ clusters for our 2-d WK-means algorithm. These clusters should represent one regime of higher correlation and one of a lower correlation.

Indeed, for $k = 2$, we obtain one correlation regime of high correlation, where the correlation of the centroid's atoms is 0.878, and one of a lower correlation, where the correlation of the centroid's atoms is 0.737. Figures 6.11(a) and 6.11(b) show our standard plots of the correlation of the centroids and log returns respectively of each cluster. Figure 6.11(c) shows the rolling correlation plot with cluster colors. We see from the plot that the second (red) cluster has caught instances of higher correlation above the average line, represented by the black horizontal line, while the first (green) cluster tends to fall below this line indicating that it captures periods of lower correlation. Figures 6.11(d) and 6.11(e) show the same plot where only one cluster is shown.

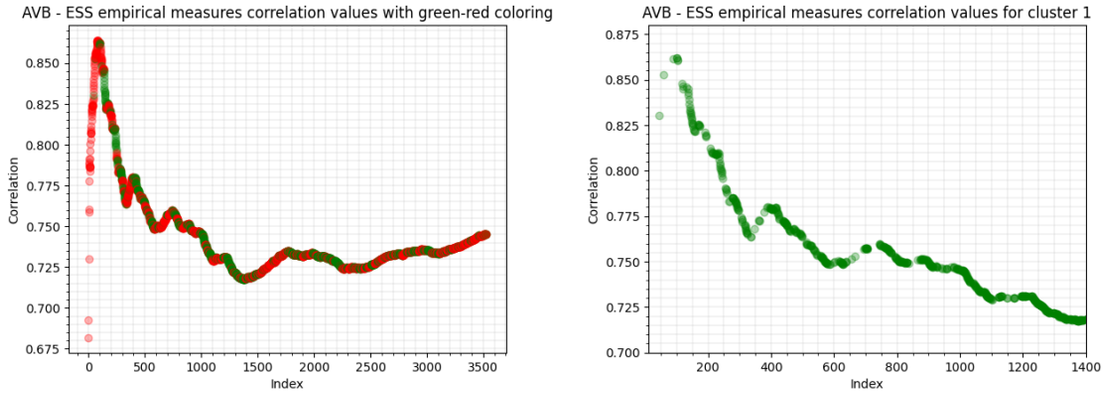


(a) AVB mean-variance clustering plot.

(b) ESS mean-variance clustering plot.

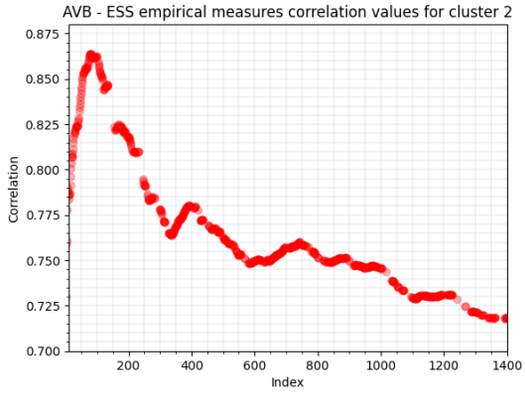
Figure 6.9: Uni-d 1-WK-means applied to AVB and ESS mean-variance clustering plots.

Figure 6.10(a) shows the correlation of each of the empirical measures that we have clustered with their associated cluster colors. We note that where the correlation is increasing the second (red) cluster tends to dominate while where the correlation is decreasing, the first (green) cluster is more apparent. These trends are further reinforced in figures 6.10(b) and 6.10(c) which show a portion of figure 6.10(a) magnified and where only one cluster is presented.



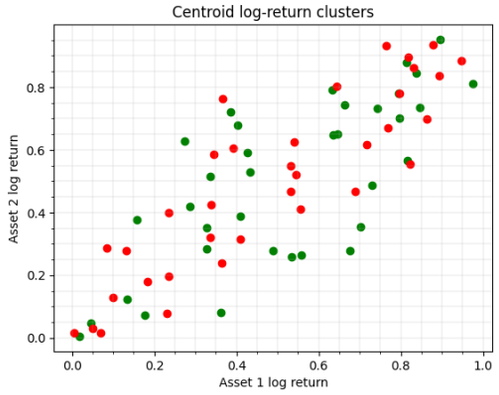
(a) AVB and ESS empirical measures correlation plot with cluster colors.

(b) AVB and ESS empirical measures correlation plot with cluster colors, cluster 1 only.

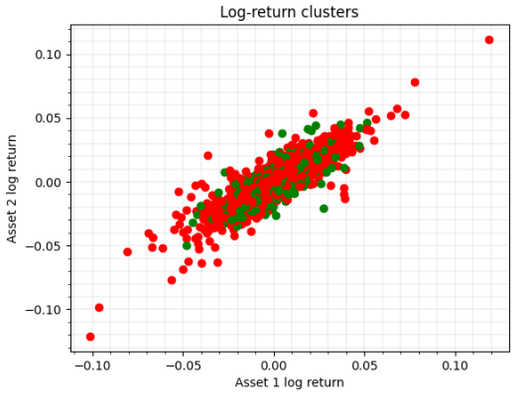


(c) AVB and ESS empirical measures correlation plot with cluster colors, cluster 2 only.

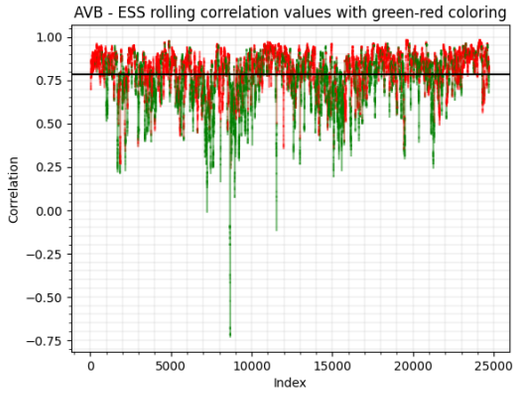
Figure 6.10: Empirical measures correlation plots of AVB and ESS for 2 clusters.



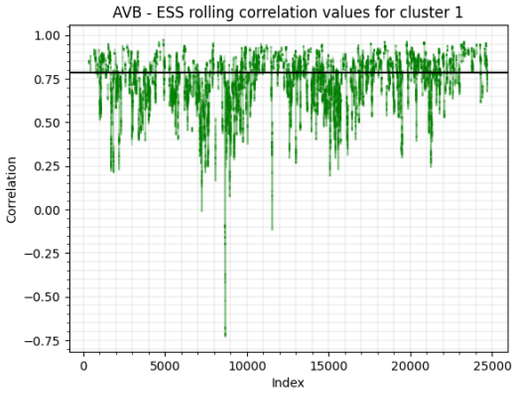
(a) Centroids 1 and 2 log-returns plot with cluster colors.



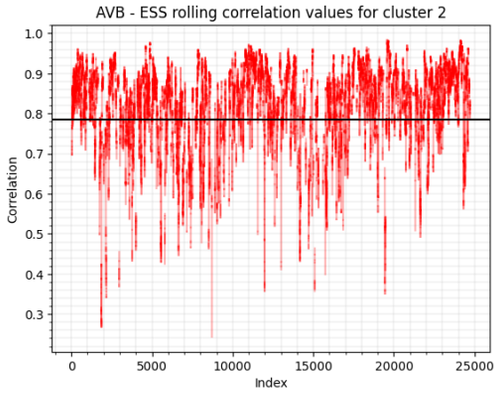
(b) Clusters 1 and 2 log-returns plot with cluster colors.



(c) AVB and ESS rolling correlation of returns plot using a rolling window of 35 market hours with cluster colors.



(d) AVB and ESS rolling correlation of returns plot using a rolling window of 35 market hours with cluster colors, cluster 1 only.

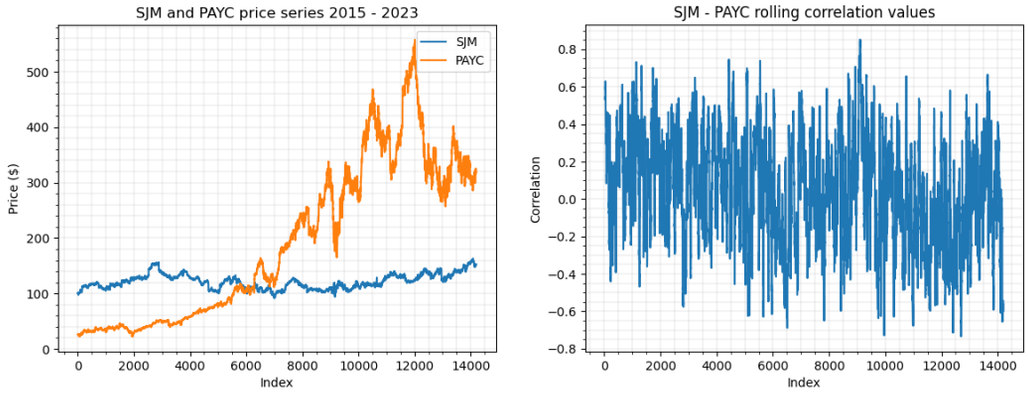


(e) AVB and ESS rolling correlation of returns plot using a rolling window of 35 market hours with cluster colors, cluster 2 only.

Figure 6.11: Centroid, cluster and rolling correlation plots of AVB and ESS for 2 clusters.

6.2.3 SJM - PAYC

Finally, SJM and PAYC constitute our last test pairing. J.M. Smucker Co (SJM) is a food and beverage manufacturing company while Paycom Software Inc (PAYC) is an online payroll and human resource technology provider. As one might suspect, the price series of the associated stocks tend to have low correlation on average. Indeed, using the hourly close prices for each stock from 2015-2023, the correlation of these assets over the entire testing period is 0.060 while the standard deviation of their rolling correlation using a window of 35 market hours is 29.25%. Figure 6.12(a) shows the price series evolution of SJM and PAYC stock between 2015 and 2023 while figure 6.12(b) shows the rolling correlation over a window of 35 market hours between these two assets during that time.



(a) SJM and PAYC price series 2015 - 2023.

(b) SJM and PAYC rolling correlation of returns plot using a rolling window of 35 market hours.

Figure 6.12: Price series and rolling correlation plots of SJM and PAYC.

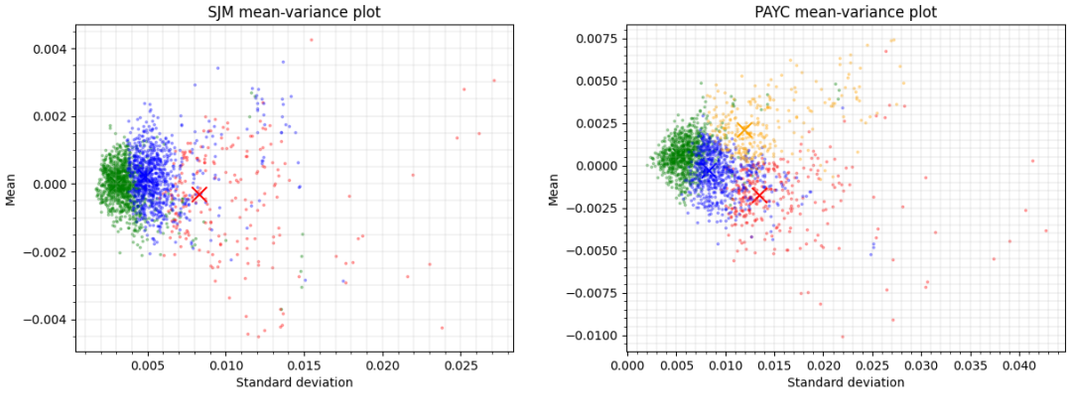
Step 1: Univariate clustering and data transformation

We initially cluster the empirical measures of each of the two assets using the uni-d 1-WK-means algorithm. Again, we begin by choosing the smallest number of clusters possible, $k = 2$ clusters, for each asset. After testing, we find that in the case of SJM and PAYC, we require 3 and 4 clusters respectively. When using $k = 3$ for SJM, the Kolmogorv-Smirnov test yields a p-value of 0.993 and when using $k = 4$ clusters for PAYC, we have a p-value of 0.999.

Figures 6.13(a) and 6.13(b) show the mean-variance clustering plots of each asset. SJM exhibits three mean-variance regimes; a low variance regime (green), a medium variance regime (blue), and a high variance regime (red). PAYC however exhibits four mean-variance regimes of increasing variance; a low variance regime (green), a medium variance regime (blue), and a higher variance regime which is further split into two regimes, one of a higher mean (orange) and one of a lower mean (red).

Step 2: Multivariate clustering

We stick to the simplest approach possible and take $k = 2$ in the 2-d WK-means algorithm. The clusters should represent one regime of higher correlation, and one regime of a lower correlation. Indeed, for $k = 2$ we obtain one correlation regime of high correlation, where the correlation of the centroid's atoms is 0.268, and one of a lower correlation, where the correlation of the centroid's atoms is -0.059. Figure 6.15(c) shows the rolling correlation plot with cluster colors. We see from the plot that the second (red) cluster has caught instances of higher correlation above the average line, represented by the black horizontal line, while the first (green) cluster tends to fall below this line indicating that it captures periods of lower correlation. Figures 6.15(d), and 6.15(e) show the same plot where only one cluster is shown.

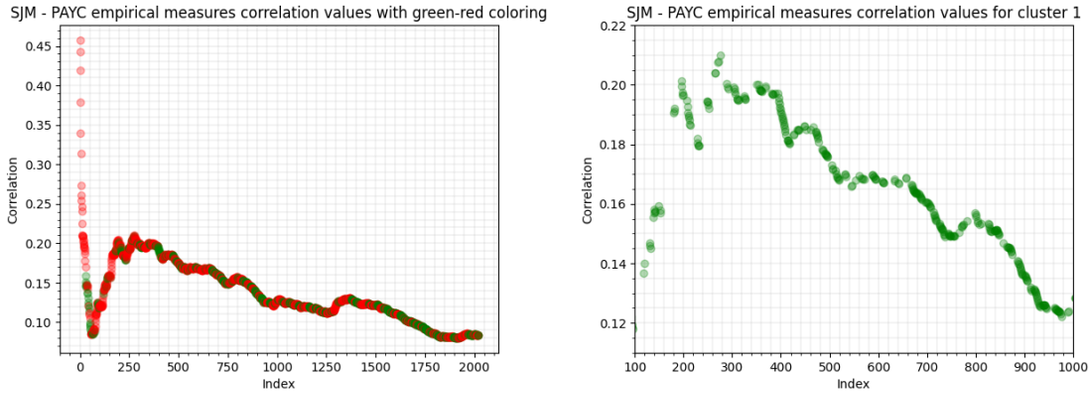


(a) SJM mean-variance clustering plot.

(b) PAYC mean-variance clustering plot.

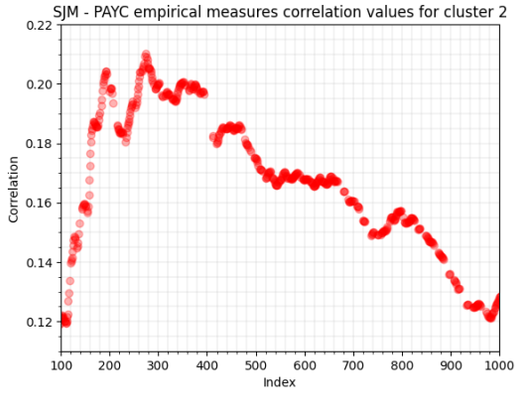
Figure 6.13: Uni-d 1-WK-means applied to SJM and PAYC mean-variance clustering plots.

Figure 6.14(a) shows the correlation of each of the empirical measures that we have clustered with their associated cluster colors. We note that as we might expect, the second (red) and first (green) clusters tend to pick up instances of stronger and weaker correlation respectively. These trends are further reinforced in figures 6.14(b), and 6.14(c) which show a portion of figure 6.14(a) magnified and where only one cluster is presented.



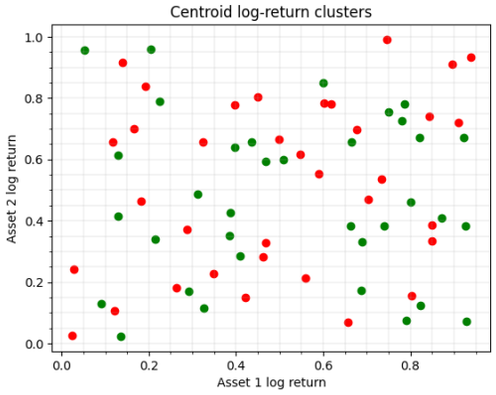
(a) AVB and ESS empirical measures correlation plot with cluster colors.

(b) SJM and PAYC empirical measures correlation plot with cluster colors, cluster 1 only.

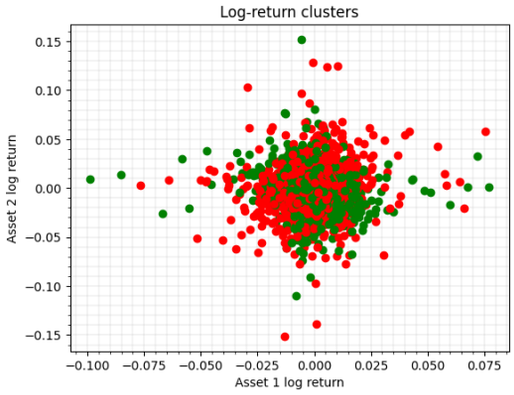


(c) SJM and PAYC empirical measures correlation plot with cluster colors, cluster 2 only.

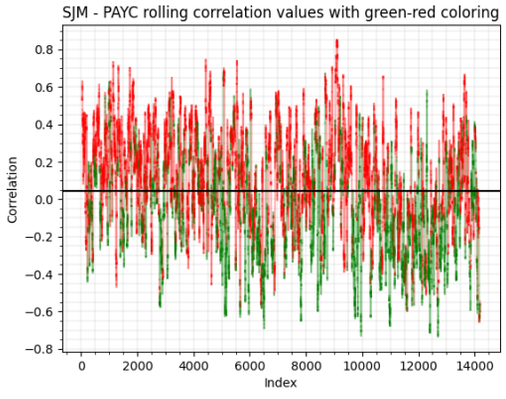
Figure 6.14: Empirical measures correlation plots of SJM and PAYC for 2 clusters.



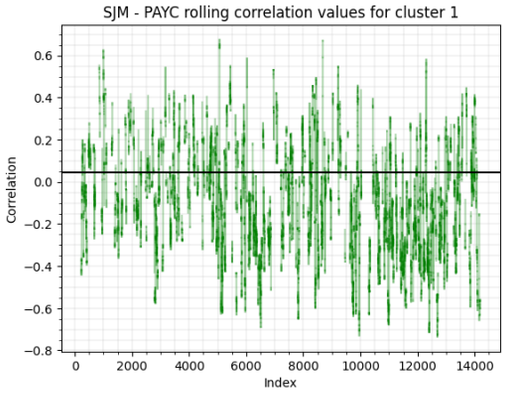
(a) Centroids 1 and 2 log-returns plot with cluster colors.



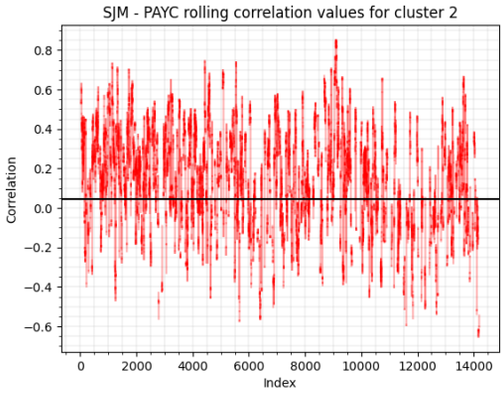
(b) Clusters 1 and 2 log-returns plot with cluster colors.



(c) SJM and PAYC rolling correlation of returns plot using a rolling window of 35 market hours with cluster colors.



(d) SJM and PAYC rolling correlation of returns plot using a rolling window of 35 market hours with cluster colors, cluster 1 only.



(e) SJM and PAYC rolling correlation of returns plot using a rolling window of 35 market hours with cluster colors, cluster 2 only.

Figure 6.15: Centroid, cluster and rolling correlation plots of SJM and PAYC for 2 clusters.

6.3 Selection of hyperparameters

Choosing values of h_1 and h_2 when using the 2-d clustering algorithm can often be more of an art than a science. The correlation between different assets may change with different levels of frequency and over different lengths of time which in turn may justify different values of (h_1, h_2) depending on the pair of assets. Although we do not know the true underlying probability distributions which exist in our real data, we can attempt to test how stable the correlation regimes we find are when we use different hyperparameters (h_1, h_2). To do so, for each pair of assets, we run the 2-d WK-means algorithm for several pairs of hyperparameters as shown in the first column of table 6.5. We do this for $n = 200$ runs. Subsequently, we then take the average percentage occurrence of the most common cluster designation for each empirical measure. For example, suppose that we choose $(h_1, h_2) = (70, 63)$ and suppose empirical measure μ_i, for some i in $\{1, \ldots, M\}$, is found in the low correlation cluster in 150 of our 200 runs. We would then say that, under the chosen hyperparameters, approximately 75% of the time our algorithm will place μ_i in the same cluster. We repeat this for all measures μ_j, for j in $\{1, \ldots, M\}$, and take a mean average of the scores. This should give us an idea as to which hyperparameter pairing gives us the greatest stability for a given pair of assets. We note that we ensure $h_1 - h_2 = 7$ throughout as we then have approximately the same number of measures M, and thus the same volume of data to cluster, for each pair of (h_1, h_2).

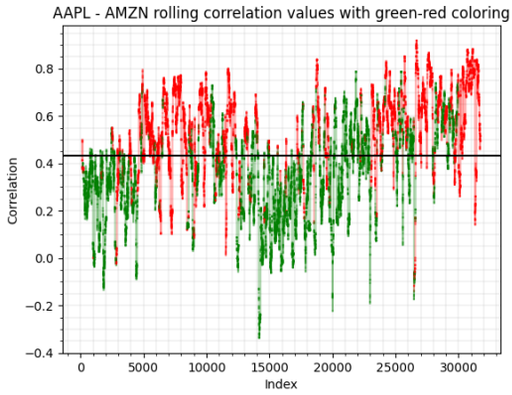
(h_1, h_2)	AAPL-AMZN	AVB-ESS	SJM-PAYC
(35, 28)	82.87%	61.67%	65.26%
(70, 63)	73.37%	69.96%	74.27%
(105, 98)	79.82%	74.32%	79.23%
(140, 133)	84.58%	74.97%	83.32%
(175, 168)	85.13%	78.15%	77.82%
(210, 203)	85.81%	82.97%	78.15%

Table 6.5: Average percentage occurrence of the most common cluster designation for each empirical measure, $n = 200$ runs.

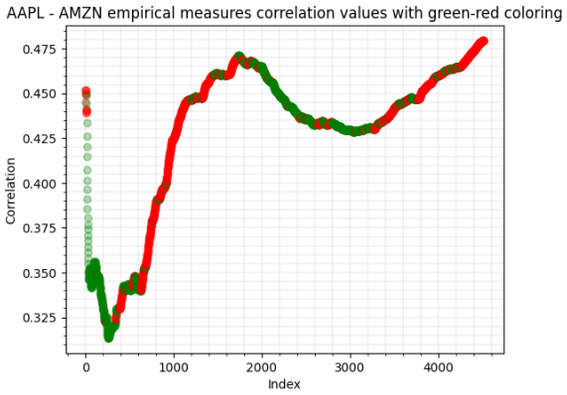
Table 6.5 shows our results. We refer to each result as a stability score. In the case of AAPL-AMZN, we note that the clustering is actually quite stable for the lowest pairing $(h_1, h_2) = (35, 28)$ but should we increase the values of h_1 and h_2 we then see a drop in stability before it recovers and peaks at approximately 84-85% with $(h_1, h_2) = (140, 133)$. The difference in stability when using $(h_1, h_2) = (35, 28)$ and $(h_1, h_2) = (140, 133)$ is less than 2% which suggests that using a h_1 value of one market week is suitable. Figure 6.16 shows the rolling correlation and measure correlation plots should we take $(h_1, h_2) = (140, 133)$. One benefit of increasing our hyperparameter values is that we lose many spurious correlations and our plots become considerably clearer, whilst maintaining the same trends. In particular, the empirical measure correlation plots show that the algorithm distinguishes between periods of increasing and decreasing correlation quite effectively.

When clustering with AVB-ESS, we see a clear trend in that increasing the (h_1, h_2) hyperparameters yields an increase in stability score. In fact, when $(h_1, h_2) = (35, 28)$ the 2-d WK-means algorithm only yields the same average clustering on approximately 60% of the runs. In comparison, should we increase (h_1, h_2) to $(210, 203)$, a h_1 value corresponding to six market weeks, then our clusters are much more stable. The lower stability for AVB-ESS when compared to AAPL-AMZN is not surprising given the volatility in the rolling correlation is much lower, making each empirical measure more susceptible to switching cluster. Figure 6.17 shows the rolling correlation and measure correlation plots when using $(h_1, h_2) = (210, 203)$. Again, our plots become clearer while maintaining the same trends and the empirical measure correlation plots show that the algorithm distinguishes between periods of increasing and decreasing correlation effectively.

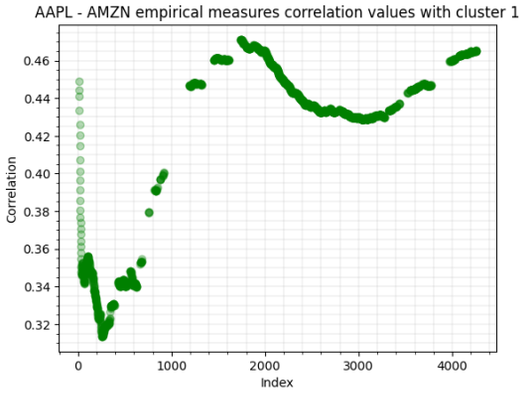
Finally, when clustering SJM-PAYC we find a similar trend to that of AVB-ESS in that the stability score tends to increase as (h_1, h_2) increases. In this case however, the score peaks at $(h_1, h_2) = (140, 133)$ and subsequently decreases. This highlights the specific nature of the hyperparameter pairs and that a 'one-size-fits-all' approach may not always yield the best results. Figure 6.18 shows the rolling correlation and measure correlation plots when using $(h_1, h_2) = (140, 133)$.



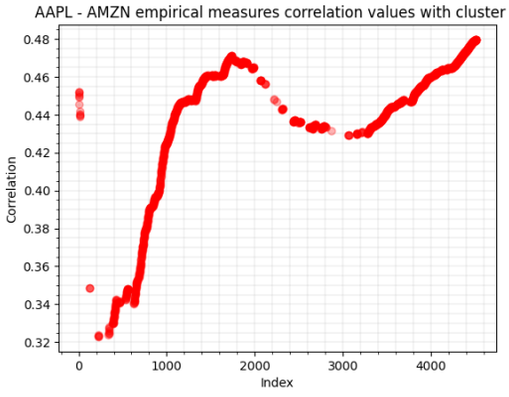
(a) AAPL and AMZN rolling correlation plot with cluster colors, $(h_1, h_2) = (140, 133)$.



(b) AAPL and AMZN empirical measures correlation plot with cluster colors, $(h_1, h_2) = (140, 133)$.

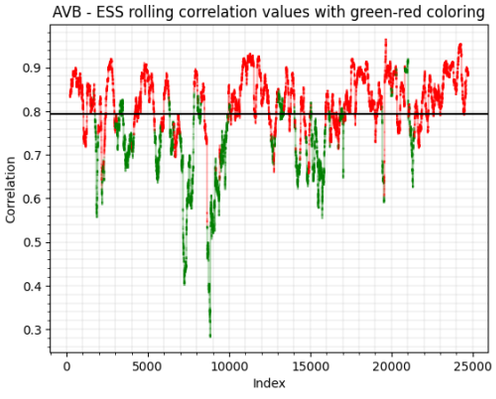


(c) AAPL and AMZN empirical measures correlation plot with cluster colors, $(h_1, h_2) = (140, 133)$ cluster 1 only.

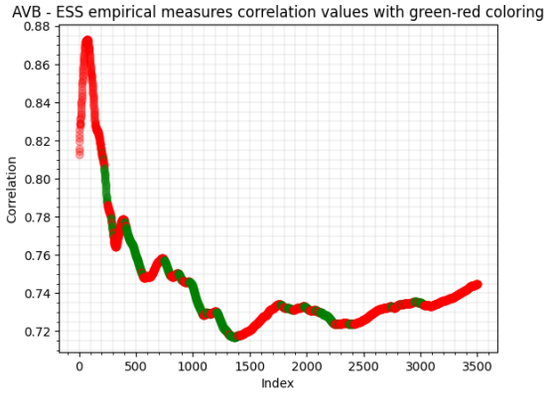


(d) AAPL and AMZN empirical measures correlation plot with cluster colors, $(h_1, h_2) = (140, 133)$ cluster 2 only.

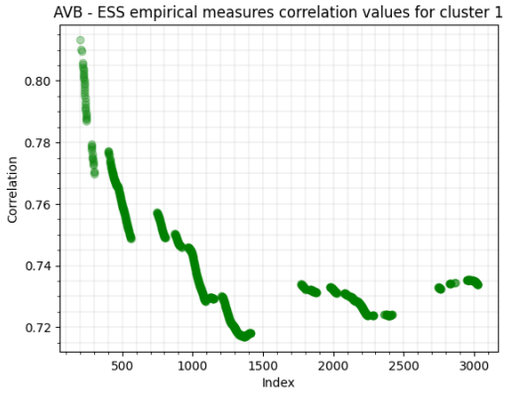
Figure 6.16: Correlation plots of AAPL and AMZN for 2 clusters, $(h_1, h_2) = (140, 133)$.



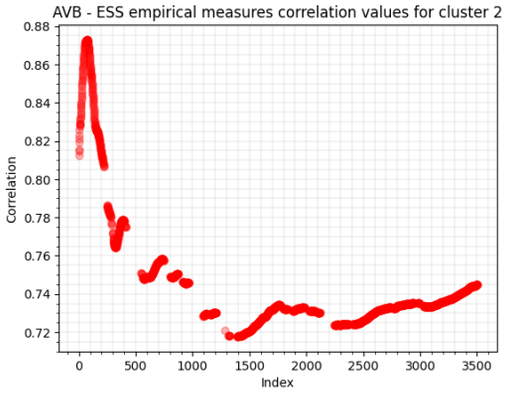
(a) AVB and ESS rolling correlation plot with cluster colors, $(h_1, h_2) = (210, 203)$.



(b) AVB and ESS empirical measures correlation plot with cluster colors, $(h_1, h_2) = (210, 203)$.

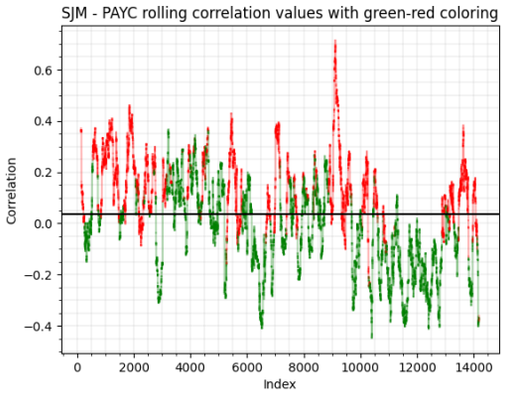


(c) AVB and ESS empirical measures correlation plot with cluster colors, $(h_1, h_2) = (210, 203)$ cluster 1 only.

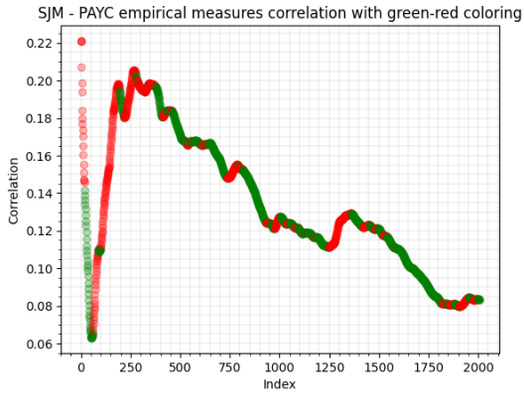


(d) AVB and ESS empirical measures correlation plot with cluster colors, $(h_1, h_2) = (210, 203)$ cluster 2 only.

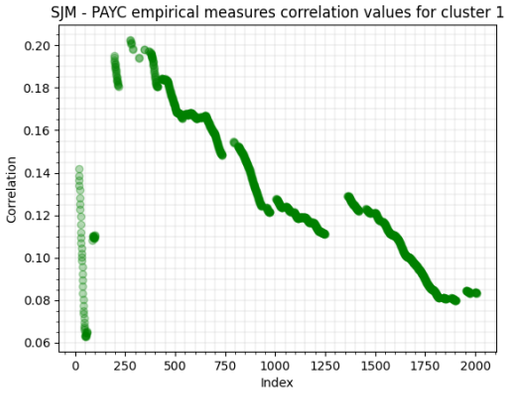
Figure 6.17: Correlation plots of AVB and ESS for 2 clusters, $(h_1, h_2) = (210, 203)$.



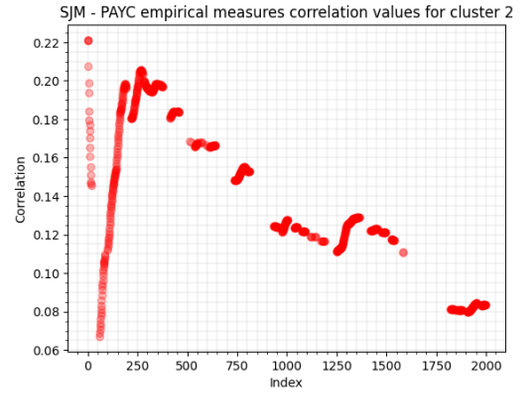
(a) SJM and PAYC rolling correlation plot with cluster colors, $(h_1, h_2) = (140, 133)$.



(b) SJM and PAYC empirical measures correlation plot with cluster colors, $(h_1, h_2) = (140, 133)$.



(c) SJM and PAYC empirical measures correlation plot with cluster colors, $(h_1, h_2) = (140, 133)$ cluster 1 only.



(d) SJM and PAYC empirical measures correlation plot with cluster colors, $(h_1, h_2) = (140, 133)$ cluster 2 only.

Figure 6.18: Correlation plots of SJM and PAYC for 2 clusters, $(h_1, h_2) = (140, 133)$.

Chapter 7

Trading strategies

In this chapter, we use our new two-step clustering-based approach to facilitate a trading strategy. Our candidate assets will be the three pairs of stocks described in chapter 6 and the overall trading strategy is based on Modern Portfolio Theory (MPT).

7.1 Modern Portfolio Theory

First proposed by H. Markowitz in 1952 [78], Modern Portfolio Theory suggests we assemble our portfolio such that we maximise our expected returns μ_p for a given level of risk σ_p. This is commonly referred to as mean-variance optimisation. For two assets, this problem may be expressed in symbolic form as

$$\begin{aligned} &\max_w \mu_p = w^T \mu \\ &\text{s.t. } \sigma_t^2 = w^T \Sigma w \text{ and } w^T \mathbf{1} = 1, \end{aligned}$$

where w is our 2×1 weights vector, μ is the 2×1 vector of expected returns, σ_t^2 is our target portfolio variance, $\mathbf{1}$ is a 2×1 vector of ones (highlighted in bold so as to distinguish it from the integer value 1), and Σ is the 2×2 covariance matrix of our constituent assets. For a portfolio of two assets one may write the covariance matrix Σ as

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho \times \sigma_1 \times \sigma_2 \\ \rho \times \sigma_1 \times \sigma_2 & \sigma_2^2 \end{pmatrix},$$

where σ_1^2 is the variance of asset 1, σ_2^2 is the variance of asset 2, and ρ is the correlation between the two assets. The constraint of $w^T \mathbf{1} = 1$ ensures the total weight of our portfolio sums to 1. This optimisation problem has a dual form where we instead minimise the variance of our portfolio for a given target return, μ_t,

$$\min_w \sigma_p^2 = w^T \Sigma w \tag{7.1.1}$$

$$\text{s.t. } \mu_t = w^T \mu \text{ and } w^T \mathbf{1} = 1. \tag{7.1.2}$$

This is a constrained minimisation problem and we may solve it using a Lagrangian function and matrix algebra.

Theorem 7.1.1 (Lagrange multipliers, [79] (Pages 282 - 289))**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be our objective function we wish to minimise and $g : \mathbb{R}^n \to \mathbb{R}^c$ be our constraints function, and let f, g both be continuous and at least once differentiable. Suppose w^* is an optimal solution to the optimisation problem*

$$\begin{aligned} &min_w \sigma_p^2 = w^T \Sigma w \\ &s.t.\ \mu_t = w^T \mu \text{ and } w^T \boldsymbol{1} = 1, \end{aligned}$$

such that rank$(D(g(w^*))) = c < n$ where $D(g(w))$ is the matrix of partial derivatives, $[D(g(w)]_{j,k} = [\frac{\partial g_j}{\partial g_k}]$. Then there exists a unique Lagrange multiplier $\lambda^* \in \mathbb{R}^c$ such that $D(f(w^*)) = \lambda^{*T} D(g(w^*))$.

By theorem 7.1.1 if there exists a local minima for equation (7.1.1) given constraints (7.1.2), then the gradient of our objective function $w^T \Sigma w$ may be expressed as a linear sum of the gradients of the constraints $\mu_t = w^T \mu$ and $w^T \mathbf{1} = 1$ where λ^* will be the coefficients. Our Lagrangian function can therefore be expressed as

$$L(w, \lambda_1, \lambda_2) = w^T \Sigma w + \lambda_1 (w^T \mu - \mu_t) + \lambda_2 (w^T \mathbf{1} - 1).$$

The first order conditions of this function are

$$\begin{aligned}
\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial w} &= 2\Sigma w + \lambda_1 \mu + \lambda_2 \mathbf{1} = 0, \\
\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_1} &= w^T \mu - \mu_t = 0, \\
\frac{\partial L(w, \lambda_1, \lambda_2)}{\partial \lambda_2} &= w^T \mathbf{1} - 1 = 0,
\end{aligned}$$

and subsequently, we may express our first order conditions as a system of linear equations using matrix algebra

$$\begin{pmatrix} 2\Sigma & \mu & \mathbf{1} \\ \mu^T & 0 & 0 \\ \mathbf{1}^T & 0 & 0 \end{pmatrix} \begin{pmatrix} w \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \mu_t \\ \mathbf{1} \end{pmatrix},$$

which is an equation of the form $Ax = b$. Should A be invertible, then it has solution $x = A^{-1}b$ where

$$A = \begin{pmatrix} 2\Sigma & \mu & \mathbf{1} \\ \mu^T & 0 & 0 \\ \mathbf{1}^T & 0 & 0 \end{pmatrix}, \quad x = \begin{pmatrix} w \\ \lambda_1 \\ \lambda_2 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ \mu_t \\ \mathbf{1} \end{pmatrix}. \tag{7.1.3}$$

In order to solve equation (7.1.3) in python, we make use of the *scipy.linalg* package and its *solve* method. Passing the matrix A and vector b as inputs, the method returns the array x. We then proceed to normalise the array such that the weights are less than 1 in absolute value.

7.2 The Magic Genie strategy

The Modern Portfolio Theory approach requires four variables: approximations of the expected returns μ, the variance σ^2, and the correlation ρ of the assets in our portfolio, and a target return μ_t. In using our two step method with the uni-d 1-WK-means and 2-d WK-means algorithms we cluster both the marginal and joint distributions into distinct clusters. These clusters each have an associated centroid which should act as an average of that cluster's constituent data. When we cluster with the uni-d 1-WK-means algorithm, our synthetic and real data experiments would lead us to expect the mean and variance of each cluster's centroid's atoms to be an average representation of the returns data in its cluster. Similarly, when clustering with the 2-d WK-means algorithm we would expect the correlation of the centroid's atoms to be representative of the returns data in its cluster. Therefore our two-step approach lends itself nicely to the mean-variance optimisation problem where we may use each cluster's centroid's mean, variance and correlation values as approximations to the true values in our calculations.

We call the following experiment the Magic Genie strategy. Suppose you rub on a magical, mathematical lamp from which a gilet-wearing genie appears. The genie can see the entire price path of each of the assets in your portfolio. However, they will not tell you what the future returns will be. Instead, at each period t, the genie will tell you which cluster the next set of returns belongs to when clustered with the uni-d 1-WK-means algorithm and 2-d WK-means algorithm as described in our two-step approach. They will also tell you the mean, variance and correlation of each cluster's centroid but that is all. Now, should the approximations garnered from our two-step approach be valid, we should hope that using each centroid's approximations for the mean, variance and correlation of the assets at a given time will lead to a profitable portfolio when substituted into equation (7.1.3). Should this not be the case, then our genie is cruel and we should throw their lamp into the ocean.

Formally, we apply our two-step approach to our three pairs of assets AAPL-AMZN, AVB-ESS, and SJM-PAYC for the periods 2005-2022, 2009-2022 and 2015-2022 respectively. We assume that we want to trade over a specific window and take $h_1 = 35$, equivalent to one market week in market hours, and $h_2 = 28$ for both the uni-d 1-WK-means and 2-d WK-means algorithms. As demonstrated in figure 5.2, each return is then associated with the first empirical measure it is included within and thus we approximate its mean, variance and correlation using the atoms of the centroid of the cluster associated to that measure. We form our mean-variance portfolios using these approximations and refer to this as the Magic Genie (MG) portfolio. In order to benchmark our strategy, we provide an alternative portfolio where we have used the average over a rolling window of 35 returns as an approximation to the mean, variance and correlation of each asset. We refer to this portfolio as the Rolling Average (RA) portfolio.

Each portfolio starts with a value of one and each trading period is one market hour. The portfolio cumulative returns are shown in table 7.1 for different values of μ_t. We see that the choice of μ_t does have an impact on the returns generated by the MG strategy but in all cases we generate a positive return. Figure 7.1 shows the performance of each pair when using its optimal target return tested. That is to say, our target return μ_t for the pairings in each respective period is 0.1%, 0.05% and 0.15% for AAPL-AMZN, AVB-ESS, and SJM-PAYC respectively. Tables 7.2, 7.3 and 7.4 show statistics for the MG and RA strategies when trading each asset pair under its optimal target return. Statistics provided include the cumulative return of each portfolio, calculated as the cumulative product of returns over the testing period when starting from a portfolio value of 1 and the classic annualised Sharpe ratio (SR_y) defined as

$$\text{SR}_y = \text{SR}_h \times \sqrt{252 \times 7} = \frac{\mu_h}{\sigma_h} \times \sqrt{252 \times 7},$$

where SR_h is the hourly Sharpe ratio, μ_h is the average hourly return adjusted to account for an annual risk free rate of 2%, and σ_h is the hourly standard deviation. We also provide the maximum drawdown of each strategy during this period, defined as being the largest drop from a peak in our cumulative returns to a subsequent trough as well as the hourly standard deviation in returns σ_h. The results generated use the hourly closing prices of each asset and do not account for market intricacies such as trading fees, bid-ask spread, shorting constraints etc.

Pair \ μ_t	0.05%	0.10%	0.15%
AAPL-AMZN	6401.37	133,781.43	8225.44
AVB-ESS	41.03	5.53	3.16
SJM-PAYC	30.55	268.39	979.85

Table 7.1: Magic Genie strategy cumulative returns for $\mu_t = 0.05\%, 0.10\%, 0.15\%$.

Strategy	Cumulative returns	Sharpe ratio	Max drawdown	σ_h
MG	133,781.43	3.03	25.55%	0.55%
RA	5.40	0.46	51.45%	0.13%

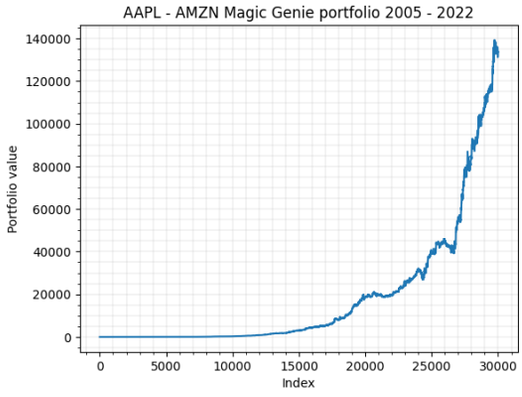
Table 7.2: Magic Genie (MG) vs Rolling Average (RA) strategy statistics for AAPL-AMZN.

Strategy	Cumulative returns	Sharpe ratio	Max drawdown	σ_h
MG	41.03	1.80	23.28%	0.37%
RA	0.92	-0.13	45.39%	0.33%

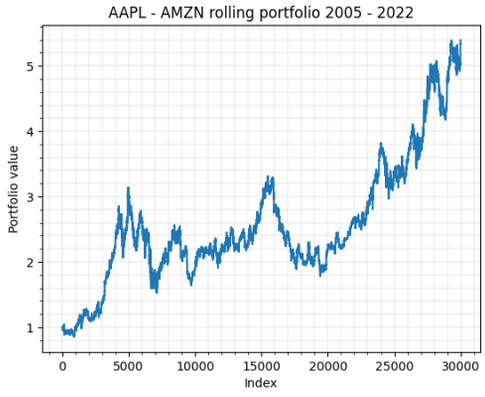
Table 7.3: Magic Genie (MG) vs Rolling Average (RA) strategy statistics for AVB-ESS.

Strategy	Cumulative returns	Sharpe ratio	Max drawdown	σ_h
MG	979.85	3.62	16.97%	0.65%
RA	2.05	0.45	28.49%	0.60%

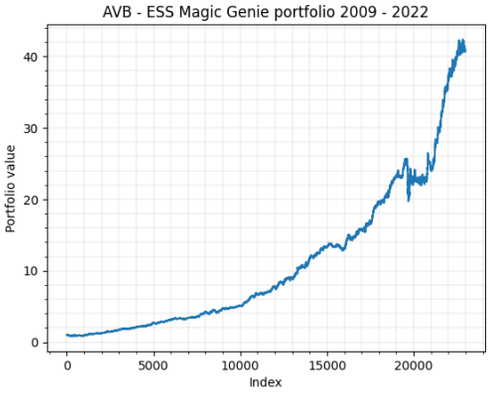
Table 7.4: Magic Genie (MG) vs Rolling Average (RA) strategy statistics for SJM-PAYC.



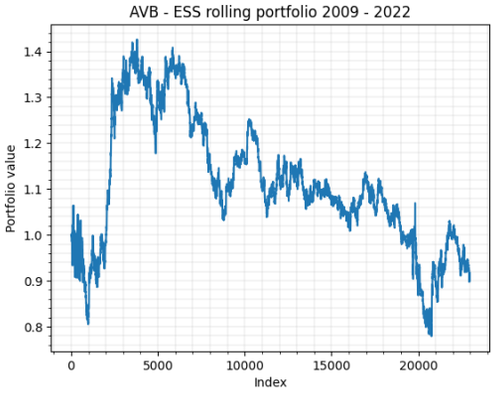
(a) AAPL - AMZN performance for the Magic Genie portfolio.



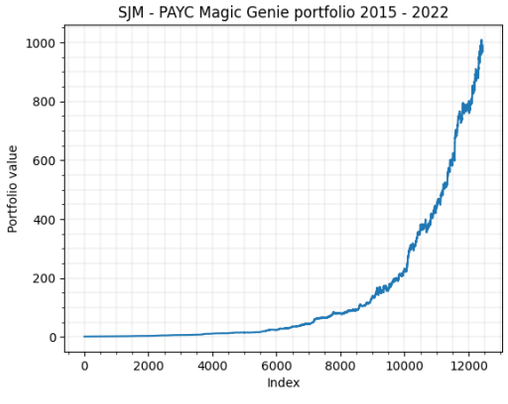
(b) AAPL - AMZN performance for the Rolling Average portfolio.



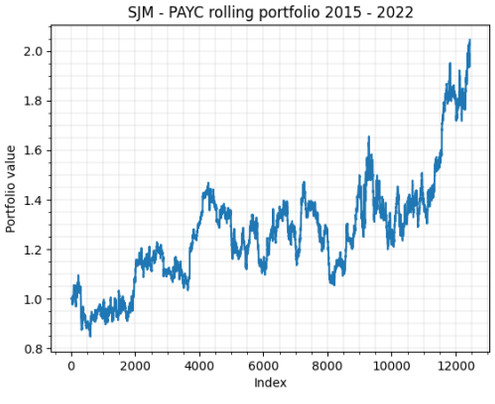
(c) AVB - ESS performance for the Magic Genie portfolio.



(d) AVB - ESS performance for the Rolling Average portfolio.



(e) SJM - PAYC performance for the Magic Genie portfolio.



(f) SJM - PAYC performance for the Rolling Average portfolio.

Figure 7.1: Magic Genie and Rolling Average portfolios.

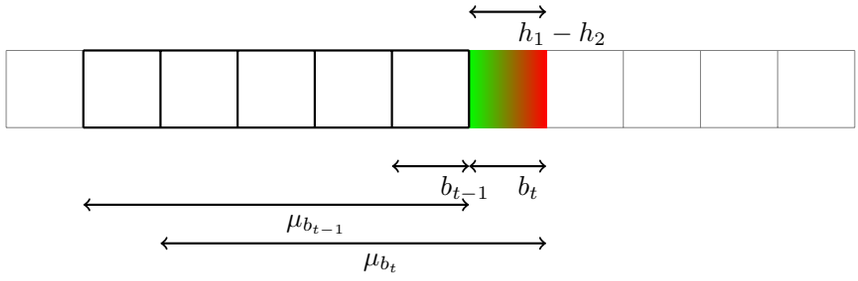


Figure 7.2: Example diagram of information used to trade with the Cluster Based (CB) strategy.

7.3 Cluster Based trading strategy

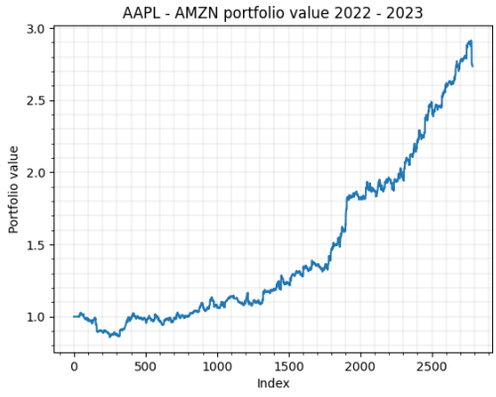
Moving beyond the realm of magic genies, we present a more realistic setting. The Magic Genie strategy makes primarily two unrealistic assumptions: that we should know the entire returns path of each asset in order to form our centroids before we begin trading, and that we may trade a given return as it occurs. We now remove those assumptions.

As in the Magic Genie experiment, we apply our two step approach to our three pairs of assets AAPL-AMZN, AVB-ESS, and SJM-PAYC for the periods 2005-2022, 2009-2022 and 2015-2022 respectively in order to obtain our clusters and centroids. We consider this to be our training data. We then use data between January 2022 and August 2023 as our testing/trading data. We therefore have removed the assumption that we should know the entire price path that we are trading. Taking $h_1 = 35$ and $h_2 = 28$ as before, we form empirical measures from our testing data and assign them to the cluster for which the p-Wasserstein distance is minimised. We do this for both the uni-d 1-WK-means clusters where we take $p = 1$ and for the 2-d WK-means clusters where we take $p = 2$. We subsequently form our mean-variance optimal portfolios using the associated cluster's centroids as described in the Magic Genie experiment and we use the same target returns as previously stipulated; that is 0.01% for AAPL-AMZN, 0.05% for AVB-ESS, and 0.15% for SJM-PAYC, given they were optimal for the training data. At no point during our testing data do we rerun our algorithm to form new centroids.

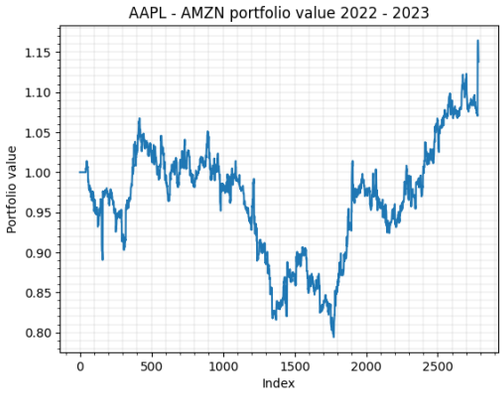
Figures 7.3(a), 7.3(c) and 7.3(e) show our results, which remain strong and profitable in each instance. We might then assume that rerunning our algorithms to form new centroids continuously is not necessary and instead, we may rerun our algorithms after a certain period of time, for example every few weeks. In doing so, we save reduce our overall computation time when running a strategy based on the clusters, as discussed in section 5.3.

We then remove our second assumption of trading a given return as it occurs, by shifting the weights of our portfolio. Each empirical measure will overlap on $h_2 = 28$ returns. Thus, after our first empirical measure, we will be able to form a new measure every $h_1 - h_2 = 7$ returns and hence we should shift our weights by 7 in order to trade. In effect, we trade on day t using information from day $t - 1$ to day $t - 5$. This is demonstrated in figure 7.2. In the MG strategy we would use information from the block of returns data forming measure μ_{b_t} in order to trade the returns in block b_t. Instead, we now use information from the block of returns data forming measure $\mu_{b_{t-1}}$ in order to trade the returns in block b_t.

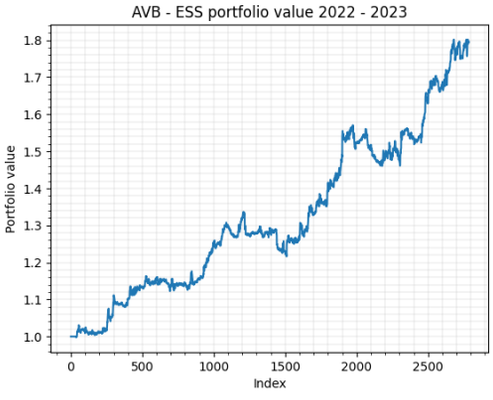
Figures 7.3(b), 7.3(d) and 7.3(f) show the performance of our Cluster Based (CB) portfolio of each pair on the testing data. Clearly the performance is markedly different to previous cases and is considerably more volatile. This shows the importance of predicting the next cluster as opposed to assuming the next cluster designation will be the same as the current one at all times.



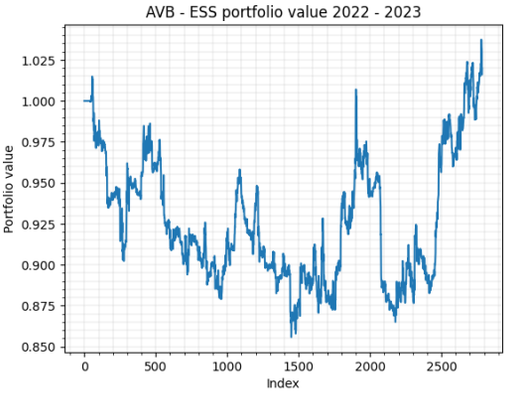
(a) AAPL - AMZN performance for the Cluster Based trading strategy, January 2022 - August 2023, using centroids from 2005 - 2022.



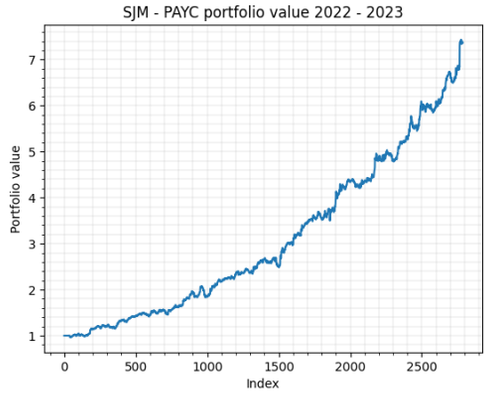
(b) AAPL - AMZN performance for the Cluster Based trading strategy, January 2022 - August 2023, using centroids from 2005 - 2022 and shifted weights.



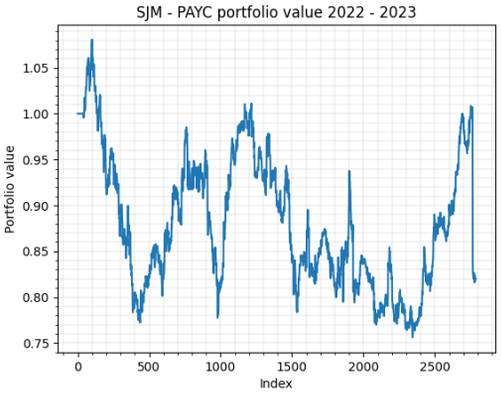
(c) AVB - ESS performance for the Cluster Based trading strategy, January 2022 - August 2023, using centroids from 2005 - 2022.



(d) AVB - ESS performance for the Cluster Based trading strategy, January 2022 - August 2023, using centroids from 2005 - 2022 and shifted weights.



(e) SJM - PAYC performance for the Cluster Based trading strategy, January 2022 - August 2023, using centroids from 2005 - 2022.



(f) SJM - PAYC performance for the Cluster Based trading strategy, January 2022 - August 2023, using centroids from 2005 - 2022 and shifted weights.

Figure 7.3: Cluster Based strategy results.

Chapter 8

Conclusion

In this thesis, we have shown that the p-Wasserstein distance and Maximum Mean Discrepancy can be combined with the k-means clustering algorithm in order to effectively partition multidimensional market returns into regimes. These regimes include periods characterised by a change in the mean and variance of the underlying distribution and periods characterised by a change in the correlation between two assets. We compared the use of each distance metric in synthetic settings and found that while the p-Wasserstein distance led to stronger clustering for correlation regimes, the MMD was more effective when clustering regimes characterised by their mean and variance changes. We then demonstrated how combining the WK-means algorithm, as described in [1], with our novel multidimensional algorithms can be effective in clustering real world market returns. Finally, we showed that knowledge of the cluster membership of each market return can be effective in building a profitable portfolio.

Areas ripe for further investigation are plentiful. These include fundamental elements to our algorithm such as our choice of clustering algorithm and distance metric in the multivariate setting. Replacing the k-means algorithm with the fuzzy c-means algorithm or a hierarchical clustering algorithm is an area of possible study. A method for determining the optimal bandwidth parameter when using the MMD would be useful while a more rigorous statistical method for choosing the number of clusters when using the 2-d WK-means algorithm to find correlation regimes in a real data environment would be preferable. Further investigation into the optimal selection of hyperparameters for a given pair of stocks when using the 2-d WK-means algorithm is also warranted. Most notably, studying the changes in joint distribution for $d > 2$ assets would be interesting. Such changes are not as simple to characterise as those of $d = 2$ but the d-dimensional WK-means and MMDK-means algorithms can be used for higher values of d. Regarding our trading strategy, we have presented only a toy model in order to illustrate the effectiveness of our clustering. A method for predicting the next cluster, possibly based on the overlapping returns or some form of traditional time series prediction, would be useful. It would also be interesting to investigate the application of mixture distributions in this context, wherein we would use a mixture of the centroid empirical distributions as opposed to only one when trading a given period of returns.

Appendix A

Technical Proofs and further results

A.1 Appendix 1

In this appendix we provide statements and proofs related to results in chapter 3 of the thesis.

A.1.1 The k-means clustering algorithm

Proposition A.1.1 (k-means clustering, [80])**.** *Given data X, the k-means algorithm produces k suitable clusters if the following is true:*

1. *There exist k clusters in the data X.*
2. *Each cluster within X is of roughly equal size.*
3. *Within-cluster variation (Definition 3.2.6) is uniform. That is, for $\delta_2 > 0$ small we have that*

$$\|WC(C_i) - WC(C_j)\| < \delta_2,$$

for $i, j = 1, \ldots, k$ and $i \neq j$.

Clusters are spherical in shape, so we expect the nearest neighbours C_j to the j^{th} centroid $\bar{x}_j$ to be contained within a ball $B(\bar{x}_j, \delta)$ where $\delta > 0$ is uniform across all clusters $j = 1, \ldots, k$. If conditions (1)-(4) are satisfied, then optimal clusterings C^* will be suitable.

Proof. See [80] for details. □

A.1.2 Maximum Mean Discrepancy

Problem 1 (Two-sample test, [51] (Problem 1))**.** Let $(\mathcal{X}, D)$ be a metric space. Suppose X and Y are independent random variables on $\mathcal{X}$. Suppose that the Borel probability measure of X is μ and that of Y is ν, where $\mu, \nu \in \mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ is the set of probability measures on $\mathcal{X}$. If we draw i.i.d. samples $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ where $x_i \sim \mu$ for $i = 1, \ldots, n$ and $y_j \sim \nu$ for $j = 1, \ldots, m$, when can we determine if $\mu \neq \nu$? In other words, we wish to implement a test for the two-sample problem

$$H_0 : \mu = \nu \quad H_1 : \mu \neq \nu.$$

Definition A.1.2 (Hilbert space, [61] (Pages 171 - 172))**.** A Hilbert space $\mathcal{H}$ is a complex vector space equipped with an inner product, that is complete with respect to the norm induced by its inner product.

Definition A.1.3 (Kernel function, [81] (Equation 3, page 3)**.** Let $\mathcal{X}$ be a topological space. We define a symmetric, similarity measure of the form

$$\begin{aligned}\kappa :&\mathcal{X} \times \mathcal{X} \to \mathbb{R} \\ &(x, x') \to \kappa(x, x'),\end{aligned}$$

such that κ returns a real number characterising the similarity between x and x'. We refer to κ as a kernel function or method.

Definition A.1.4 (Universal reproducing kernel Hilbert space, [51] (Section 2.2, page 727))**.** Let $\mathcal{X}$ be a topological space, $\mathcal{H}$ be a Hilbert space and $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a reproducing kernel. We say that the reproducing kernel Hilbert space $(\mathcal{H}, \kappa)$ is universal if $\kappa(\cdot, \cdot)$ is continuous and $\mathcal{H}$ is dense in $C(\mathcal{X})$, the space of bounded, continuous functions on $\mathcal{X}$, with respect to the L_∞ norm.

Definition A.1.5 (Characteristic kernel, [52] (Section 2))**.** Let $\mathcal{X}$ be a non-empty set. A kernel κ on $\mathcal{X}$ is called characteristic if the mean mapping

$$\mu \to \mathbb{E}_{X\sim\mu}[\kappa(\cdot, X)]$$

is injective.

A.1.3 p-Wasserstein distance

Proposition A.1.6 (The p-Wasserstein distance is a metric, [56] (Chapter 5.1, proposition 5.4))**.** *The distance $\mathcal{W}_p : \mathcal{P}_p(X) \times \mathcal{P}_p(X) \to [0, \infty)$ is a metric on $\mathcal{P}_p(X)$.*

Proof. As in [56], we show that the first three criteria of being a metric are met. The final criterion, the triangle inequality, can be shown using the 'gluing lemma' as described in [56] chapter 5.1, proposition 5.4. We note that by definition $\mathcal{W}_p(\mu, \nu) \geq 0$ for any $\mu, \nu \in \mathcal{P}_p(X)$, giving us the first requirement. Furthermore, since $D(x, y)$ is itself a metric, we must have that it is symmetrical and since $\mathbb{P} \in \Pi(\mu, \nu)$ if and only if $S\#\mathbb{P} \in \Pi(\mu, \nu)$ where $S(x, y) = (y, x)$ we have that $\mathcal{W}_p(\mu, \nu)$ is symmetrical.

Finally, if $\mu = \nu$ then we may take $\mathbb{P}(x, y) = \delta_x(y)\mu(x)$ such that

$$\mathcal{W}_p(\mu, \nu) \leq \int_{X\times X} D(x, y)^p \mathbb{P}(dx, dy) = 0,$$

as $x = y$ $\mathbb{P}$-almost everywhere. Suppose we have that $\mathcal{W}_p(\mu, \nu) = 0$. Then there must exist some $\mathbb{P} \in \Pi(\mu, \nu)$ such that $x = y$ $\mathbb{P}$-almost everywhere. Hence for any function $f : X \to \mathbb{R}$,

$$\int_X f(x)\mu(dx) = \int_{X\times X} f(x)\mathbb{P}(dx, dy) = \int_{X\times X} f(y)\mathbb{P}(dx, dy) = \int_X f(y)\nu(dy).$$

Since this holds for all f we have that $\mu = \nu$. □

Remark A.1.7. The p-Wasserstein distance, defined in definition 3.4.1, is in fact a special case of an integral probability metric via its dual formulation. As shown in [82], where $p = 1$, the dual formulation is given by

$$\mathcal{W}_1(\mu, \nu) = \sup_{f\in \mathrm{Lip}_1(X)} \left(\int_X f(x)d\mu(x) - \int_X f(y)d\nu(y) \right), \tag{A.1.1}$$

where $\mathrm{Lip}_1(X)$ denotes a collection of Lipschitz functions with Lipschitz constant 1

$$\mathrm{Lip}_1 := \{f : \|f\|_{\mathrm{Lip}} \leq 1\},$$

and where

$$\|f\|_{\mathrm{Lip}} := \sup_{x\in\mathrm{supp}(\mu), y\in\mathrm{supp}(\nu), x\neq y} \frac{|f(x) - f(y)|}{D(x, y)}.$$

Since μ and ν are probability measures we may write equation (A.1.1) as

$$\mathcal{W}_1(\mu,\nu) = \sup_{\mathrm{Lip}_1}(\mathbb{E}_\mu[f(x)] - \mathbb{E}_\nu[f(y)]).$$

Therefore the 1-Wasserstein distance is an integrable probability metric over $\mathcal{F}$ given by the unit ball in the space of functions

$$\mathrm{Lip}(X) = \{f : X \to \mathbb{R} : f \text{ continuous}, \quad \|f\|_{\mathrm{Lip}} < \infty\},$$

and it shares a relationship with the Maximum Mean Discrepancy.

A.2 Appendix 2

In this appendix we provide statements and proofs related to results in chapter 4 of the thesis.

Proposition A.2.1 ([58] (Equation (3), page 2)). *Suppose $\mu, \nu \in \mathcal{P}_p(\mathbb{R})$ and that μ, ν are absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}$. Then the p-Wasserstein distance $\mathcal{W}_p(\mu,\nu)$ is given by*

$$\mathcal{W}_p(\mu,\nu) = \left(\int_0^1 D(F_\mu^{-1}(z), F_\nu^{-1}(z))^p dz\right)^{\frac{1}{p}},$$

where the quantile function $F_\mu^{-1} : [0,1) \to \mathbb{R}$ is defined as

$$F_\mu^{-1}(z) := \inf\{x : F_\mu(x) > z\},$$

and $D(x,y) = |x-y|$.

Proof. A consequence of the fact that the unique optimal transport map pushing μ onto ν is given by $f(\mathrm{x}) = (F_\nu^{-1} \circ F_\mu)(\mathrm{x})$, and therefore

$$\mathcal{W}_p(\mu,\nu) = \left(\int_{\mathbb{R}} D(x, F_\nu^{-1}(F_\nu(x)))^p dx\right)^{\frac{1}{p}} = \left(\int_0^1 D(F_\mu^{-1}(z), F_\nu^{-1}(z))^p dz\right)^{\frac{1}{p}}$$

by a change of variables. □

Proposition A.2.2 (Uni-dimensional p-Wasserstein barycentre, [1] (Proposition 2.6, page 11)). *Suppose that $\{\mu_i\}_{1\le i\le M}$ are a family of empirical probability measures, where each measure has N atoms, $(\alpha_j^i)_{1\le j\le N} \subset \mathbb{R}^N$. Let*

$$a_j = \mathit{Median}(\alpha_j^1, \ldots, \alpha_j^M), \quad b_j = \mathit{Mean}(\alpha_j^1, \ldots, \alpha_j^M),$$

for $j = 1, \ldots, 35$. Then the cumulative distribution function of the 1-Wasserstein barycentre $\bar{\mu} \in \mathcal{P}_1(\mathbb{R})$ over $\{\mu_i\}_{1\le i\le M}$ with respect to the 1-Wasserstein distance is given by

$$\bar{\mu}((-\infty, x]) = \frac{1}{N}\sum_{i=1}^N \mathbb{1}_{\{a_i \le x\}}(x), \tag{A.2.1}$$

and the p-Wasserstein barycentre $\bar{\mu} \in \mathcal{P}_p(\mathbb{R})$ over $\{\mu_i\}_{1\le i\le M}$ with respect to the p-Wasserstein distance, for $p > 1$, is given by

$$\bar{\mu}((-\infty, x]) = \frac{1}{N}\sum_{i=1}^N \mathbb{1}_{\{b_i \le x\}}(x). \tag{A.2.2}$$

Moreover, $\bar{\mu}$ is not necessarily unique.

Proof. This proof is taken from appendix C.1 in [1], pages 33 - 34. We prove that equation (A.2.1) holds. Assume that $N = 1$, such that each measure μ_i is comprised of only one atom α_i for $i = 1, \ldots, M$. Without loss of generality we may also assume that the sequence $(\alpha_i)_{i=1}^M$ is non-decreasing. By convexity of the function $\phi_a(x) = |x - a|$ for $a \in \mathbb{R}$, the Wasserstein barycentre

will also have $N = 1$ atoms. Then, using statement (4.1.3) with $D(x, y) = |\alpha_i - \beta_i|$, the problem of finding the barycentre $\bar{\mu}$ is equivalent to the optimisation

$$\inf_{\nu \in \mathcal{P}_1(\mathbb{R})} \sum_{i=1}^{M} W_1(\mu_i, \nu) = \inf_{a \in \mathbb{R}} \sum_{i=1}^{M} |a - \alpha_i| = \inf_{a \in \mathbb{R}} \sum_{i=1}^{M} \phi_{\alpha_i(a)}. \tag{A.2.3}$$

The minimiser $a^* \in \mathbb{R}$ to the right hand side of (A.2.3) is obtained using first order conditions, where we solve $\frac{df}{dx}(x) = 0$ over $\mathbb{R}$, where

$$f(x) = |x - \alpha_1| + \cdots + |x - \alpha_m| = \phi_{\alpha_1}(x) + \cdots + \phi_{\alpha_M}(x).$$

Since

$$\frac{d\phi_{\alpha_i}}{dx}(x) = \text{sgn}(x - \alpha_i),$$

for $i = 1, \ldots, M$, we have that

$$a^* = \text{arginf}_{a \in \mathbb{R}} \sum_{i=1}^{M} |a - \alpha_i| = \text{Median}(\alpha_1, \ldots, \alpha_M). \tag{A.2.4}$$

In particular, if $M \bmod 1 = 0$, then $a^* \in [\alpha_{\frac{M}{2}}, \alpha_{\frac{M}{2}+1})$. If $M \bmod 2 = 1$, then the (unique) optimiser is given by $a^* = \alpha_K$ where $K = \lfloor \frac{M}{2} \rfloor + 1$. Setting $a = a^*$ gives (A.2.1). If $N > 1$, then the problem of finding the Wasserstein barycentre is equivalent to

$$\inf_{\nu \in \mathcal{P}_1(\mathbb{R})} \sum_{i=1}^{M} W_1(\mu_i, \nu) = \inf_{(a_1, \ldots, a_N) \in \mathbb{R}^N} \sum_{i=1}^{M} \sum_{j=1}^{N} |a_j - \alpha_i^j|.$$

Interchanging the order of summation, we have that

$$\inf_{(a_1, \ldots, a_N) \in \mathbb{R}^N} \sum_{i=1}^{M} \sum_{j=1}^{N} |a_j - \alpha_i^j| = \sum_{j=1}^{N} \left(\inf_{a_j \in \mathbb{R}} \sum_{i=1}^{M} |a_j - \alpha_i^j| \right).$$

By applying (A.2.4) to each summation over M, we obtain the desired result (A.2.1). When $p > 1$, the proof for result (A.2.2) is similar. □

A.3 Appendix 3

In this appendix we provide further results related to the experiments in chapter 5 of the thesis. We also provide a derivation of the correlated Merton jump diffusion processes used in section 5.2.

A.3.1 Geometric Brownian motion

5.1.1: Further results

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	88.12% ± 4.48%	83.53% ± 3.88%	89.44% ± 4.96%
2-d MMDK-means	95.02% ± 1.70%	95.31% ± 0.35%	94.70% ± 2.32%

Table A.1: Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = 0.5$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	82.52% ± 4.21%	87.99% ± 3.11%	80.51% ± 4.99%
2-d MMDK-means	77.17% ± 5.43%	85.42% ± 6.27%	74.24% ± 5.84%

Table A.2: Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = -0.5$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	85.89% ± 4.25%	86.25% ± 2.68%	85.57% ± 5.09%
2-d MMDK-means	74.85% ± 3.69%	87.23% ± 2.91%	70.55% ± 4.32%

Table A.3: Accuracy scores with 95% CI, GBM synthetic path with simultaneous mean-variance regimes and fixed $\rho = -1$, $n = 50$ runs.

5.1.2: Further results

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	82.28% ± 6.78%	84.07% ± 8.43%	81.49% ± 7.01%
2-d MMDK-means	88.32% ± 5.55%	84.42% ± 7.84%	89.41% ± 5.23%

Table A.4: Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 1$, $\rho_1 = 0$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	98.60% ± 1.69%	97.83% ± 1.93%	98.63% ± 1.60%
2-d MMDK-means	95.03% ± 4.20%	95.18% ± 3.80%	94.76% ± 4.32%

Table A.5: Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = -1$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	85.21% ± 6.00%	91.17% ± 5.31%	83.02% ± 6.82%
2-d MMDK-means	80.01% ± 6.67%	83.91% ± 5.63%	78.52% ± 7.26%

Table A.6: Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = -1$, $\rho_1 = 0$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	94.70% ± 3.84%	95.92% ± 3.23%	94.08% ± 4.47%
2-d MMDK-means	89.49% ± 5.59%	91.55% ± 4.94%	88.59% ± 6.03%

Table A.7: Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = -0.5$, $\rho_1 = -1$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	65.92% ± 4.31%	80.12% ± 5.00%	61.04% ± 5.93%
2-d MMDK-means	52.75% ± 2.70%	59.65% ± 2.14%	50.33% ± 3.73%

Table A.8: Accuracy scores with 95% CI, GBM synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = 0.5$, $n = 50$ runs.

A.3.2 Correlation of Merton jump diffusion processes

The following work has been adapted from [75].

Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, \mathbb{P})$ be a probability space and let $W_t = (W_t^1, W_t^2)$ be a bivariate correlated Brownian motion process adapted to the filtration. We define a covariance matrix

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix},$$

where $dW_t^1, W_t^2 = \rho dt$ with ρ the instantaneous correlation between the two Brownian motion components. We define a homogeneous Poisson counting measure $p(dy, dt) \equiv p(dy^1, dy^2, dt)$ defined

over $\mathbb{R}^2 \times [0,T]$ which is associated with a marked point process $((\mathbf{Y}_n), N_t)$. The intensity of the Poisson measure is $\lambda m_{\mathbb{P}}(dy)dt$, where $\lambda > 0$ is the constant arrival rate of the jumps of the Poisson process N_t under $\mathbb{P}$ and $m_{\mathbb{P}}(dy)$ is the probability distribution on the independently and identically distributed marks $\mathbf{Y}_n$, which is also independent of N_t and W_t.

We denote the compensated measure as

$$\hat{p}(dy, dt) = p(dy, dt) - \lambda m_{\mathbb{P}}(dy)dt.$$

We assume that we have two assets S^1 and S^2 whose return dynamics under the market measure $\mathbb{P}$ are given by

$$\frac{dS_t^i}{S_{t-}^i} = \mu^i dt + \sigma^i dW_t^i + \int_{\mathbb{R}^2} [e^{y^i} - 1]\hat{p}(dy, dt),$$

for $i = 1, 2$ where μ^i is the expected return per unit time, and σ^i is the instantaneous volatility per unit time. Since we have two assets in our model, the jump-size $\mathbf{Y} = (Y^1, Y^2)^T$ is a bivariate process taking values $y = (y^1, y^2)^T$ in $\mathbb{R}^2$. When restricting the Poisson measure to one of the jump-size components, we write

$$p(dy^i, dt) \equiv \int_{\mathbb{R}} p(dy^i, dy^j, dt)dy^j,$$

for $i = 1, 2$ and $j = 2, 1$ with the associated compensated measures

$$\hat{p}(dy^i, dt) = p(dy^i, dt) - \lambda m_{\mathbb{P}}(dy^i)dt,$$

for $i = 1, 2$ where $m_{\mathbb{P}}(dy^i)$ denotes the marginal distribution of jump-sizes Y_n^i under $\mathbb{P}$.

The Y^i for $i = 1, 2$ are random jump-sizes assumed to be correlated pairwise with correlation $\text{Corr}[Y^1, Y^2] = \rho_Y$. For our purposes we assume $\rho_Y = \rho$. We define the expected proportional jump-size as

$$\kappa^i \equiv \mathbb{E}_{\mathbb{P}}[e^{Y^i} - 1] \equiv \int_{\mathbb{R}} [e^{y^i} - 1] m_{\mathbb{P}}(dy^i).$$

We note that in terms of the compensated Poisson measure associated with each particular asset, we may write

$$\begin{aligned}
\frac{dS_t^i}{S_{t-}^i} &= \mu^i dt + \sigma^i dW_t^i + \int_{\mathbb{R}} [e^{y^i} - 1]\hat{p}(dy^i, dt) \\
&= \mu^i dt + \sigma^i dW_t^i + \int_{\mathbb{R}} [e^{y^i} - 1](p(dy^i, dt) - \lambda m_{\mathbb{P}}(dy^i)dt) \\
&= \mu^i dt + \sigma^i dW_t^i + \int_{\mathbb{R}} [e^{y^i} - 1]p(dy^i, dt) - \lambda \int_{\mathbb{R}} [e^{y^i} - 1] m_{\mathbb{P}}(dy^i)dt \\
&= (\mu^i - \lambda\kappa^i)dt + \sigma^i dW_t^i + \int_{\mathbb{R}^2} [e^{y^i} - 1]p(dy^i, dt),
\end{aligned}$$

which yields a solution of the form

$$S_t^i = S_0^i \exp\left[\left(\mu^i - \lambda\kappa^i - \frac{\sigma^i}{2}\right)t + \sigma^i W_t^i + \sum_{n=1}^{N_t} Y_n^i\right],$$

for $i = 1, 2$.

A.3.3 Merton jump diffusion process

5.2.1: Further results

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	87.54% ± 3.96%	77.76% ± 9.27%	90.59% ± 4.81%
2-d MMDK-means	97.27% ± 1.97%	96.49% ± 0.81%	97.30% ± 2.40%

Table A.9: Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and fixed $\rho = 0.5$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	90.84% ± 3.05%	78.15% ± 9.37%	94.85% ± 2.64%
2-d MMDK-means	90.71% ± 3.67%	98.46% ± 0.15%	87.92% ± 4.92%

Table A.10: Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and fixed $\rho = -0.5$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	89.05% ± 3.81%	76.12% ± 10.65%	94.97% ± 3.50%
2-d MMDK-means	88.04% ± 4.35%	95.13% ± 1.85%	85.47% ± 5.40%

Table A.11: Accuracy scores with 95% CI, MJD synthetic path with simultaneous mean-variance regimes and fixed $\rho = -1$, $n = 50$ runs.

5.2.2: Further results

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	83.41% ± 6.11%	78.35% ± 9.70%	84.90% ± 6.45%
2-d MMDK-means	77.41% ± 6.79%	79.48% ± 6.46%	76.54% ± 7.20%

Table A.12: Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 1$, $\rho_1 = 0$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	90.05% ± 4.87%	84.63% ± 9.41%	91.64% ± 6.11%
2-d MMDK-means	90.98% ± 5.12%	84.26% ± 9.39%	93.00% ± 4.10%

Table A.13: Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = -1$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	82.95% ± 6.56%	74.09% ± 10.39%	85.71% ± 6.77%
2-d MMDK-means	75.77% ± 6.75%	79.08% ± 7.06%	74.94% ± 7.33%

Table A.14: Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = -1$, $\rho_1 = 0$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	82.09% ± 5.58%	84.48% ± 7.59%	81.10% ± 6.91%
2-d MMDK-means	89.34% ± 5.54%	88.48% ± 6.61%	89.43% ± 5.39%

Table A.15: Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = -0.5$, $\rho_1 = -1$, $n = 50$ runs.

Algorithm	Total	Regime-on	Regime-off
2-d WK-means	64.86% ± 4.63%	77.81% ± 7.73%	60.40% ± 6.94%
2-d MMDK-means	55.86% ± 2.54%	66.40% ± 4.73%	52.23% ± 2.33%

Table A.16: Accuracy scores with 95% CI, MJD synthetic path with simultaneous correlation regimes and $\rho_0 = 0$, $\rho_1 = 0.5$, $n = 50$ runs.

A.4 Appendix 4

In this appendix we provide statements and proofs related to results in chapter 6 of the thesis.

Proposition A.4.1 (Probability transformation)**.** *Suppose X is a random variable with continuous CDF F. Then $F(X) \sim$ Uniform(0, 1).*

Proof. Let X be a continuous random variable with CDF $F(x)$. We define $Y := F(X)$ such that $G(y)$ is the CDF of Y. Then for $y \in [0, 1]$, if $F^{-1}(y)$ exists we have that

$$\begin{aligned} G(y) &= \mathbb{P}(Y \leq y) \\ &= \mathbb{P}(F(X) \leq y) \\ &= \mathbb{P}(X \leq F^{-1}(y)) \\ &= F(F^{-1}(y)) \\ &= y \end{aligned}$$

If $F^{-1}(y)$ does not exist then we replace it with the generalised inverse $F^{\leftarrow}(y) = \inf\{x : F(x) \geq y\}$ for $y \in (0, 1)$, $F^{\leftarrow}(-\infty) = 0$ and $F^{\leftarrow}(\infty) = 1$, and the result still holds. □

Bibliography

[1] Blanka Horvath, Zacharia Issa, and Aitor Muguruza. Clustering market regimes using the wasserstein distance. *arXiv preprint arXiv:2110.11848*, 2021.

[2] Yingjie Tian and Dongkuan Xu. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2:165–193, 2015.

[3] Pierpaolo D'Urso and Elizabeth Ann Maharaj. Wavelets-based clustering of multivariate time series. *Fuzzy Sets and Systems*, 193:33–61, 2012.

[4] T Warren Liao. Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874, 2005.

[5] J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297. University of California Los Angeles LA USA, 1967.

[6] Arbey Aragón, Andrés Arévalo, Germán Hernández, Diego León, Javier Sandoval, and Jaime Niño. Clustering algorithms for risk-adjusted portfolio construction. *Procedia Computer Science*, 108:1334–1343, 2017.

[7] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[8] Hailin Li. Multivariate time series clustering based on common principal component analysis. *Neurocomputing*, 349:239–247, 2019.

[9] Ammar Belatreche, Ahmed Bouridane, Baqar Rizvi, and Ian Watson. Detection of stock price manipulation using kernel based principal component analysis and multivariate density estimation. *IEEE Access*, 8:135989–136003, 2020.

[10] Weiyun Huang, Erik Johnson, Hillol Kargupta, and Krishnamoorthy Sivakumar. Distributed clustering using collective principal component analysis. *Knowledge and Information Systems*, 3:422–448, 2001.

[11] Miriam-Webster. The history of 'bull' and 'bear' markets.

[12] Frank J Fabozzi and Jack Clark Francis. Stability tests for alphas and betas over bull and bear market conditions. *The Journal of Finance*, 32(4):1093–1099, 1977.

[13] Marcelle Chauvet and Simon Potter. Coincident and leading indicators of the stock market. *Journal of Empirical Finance*, 7(1):87–111, 2000.

[14] Blanka Horvath and Zacharia Issa. Non-parametric online market regime detection and regime clustering for multidimensional and path-dependent data structures. *arXiv preprint arXiv:2306.15835*, 2023.

[15] John Powell, Rubén Roa, Jing Shi, and Viliphonh Xayavong. A test for long-term cyclical clustering of stock market regimes. *Australian Journal of Management*, 32(2):205–221, 2007.

[16] John M Maheu, Thomas H McCurdy, and Yong Song. Components of bull and bear markets: bull corrections and bear rallies. *Journal of Business & Economic Statistics*, 30(3):391–403, 2012.

[17] Massimo Guidolin and Allan Timmermann. Economic implications of bull and bear regimes in uk stock and bond returns. *The Economic Journal*, 115(500):111–143, 2005.

[18] Ning Hao, Yue S Niu, and Heping Zhang. Multiple change-point detection: a selective overview. *Statistical Science*, pages 611–623, 2016.

[19] Nicolas Chopin. Dynamic detection of change points in long time series. *Annals of the Institute of Statistical Mathematics*, 59(2):349, 2007.

[20] SR Idate and Harshada C Mandhare. A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 931–935. IEEE, 2017.

[21] Imanol Perez Arribas, Thomas Cochrane, Peter Foster, and Terry Lyons. Anomaly detection on streamed data. *arXiv preprint arXiv:2006.03487*, 2020.

[22] James D Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the econometric society*, pages 357–384, 1989.

[23] Theis Lange and Anders Rahbek. An introduction to regime switching time series models. In *Handbook of Financial Time Series*, pages 871–887. Springer, 2009.

[24] Nicolas Champagnat, Madalina Deaconu, Antoine Lejay, Nicolas Navet, and Khaled Salhi. Regime switching model for financial data: Empirical risk analysis. *Physica A: Statistical Mechanics and its Applications*, 461:148–157, 2016.

[25] Jun Chen and Edward PK Tsang. Constructing a bellwether theory: Regime change detection using directional change. In *2017 9th Computer Science and Electronic Engineering (CEEC)*, pages 112–115. IEEE, 2017.

[26] A Taylan Cemgil, Fikret S Gürgen, Nesrin Okay, and M Serdar Yümlü. Bayesian changepoint and time-varying parameter learning in regime switching volatility models. *Digital Signal Processing*, 40:198–212, 2015.

[27] Rongbo Chen, Jean-Marc Patenaude, Mingxuan Sun, Shengrui Wang, and Kunpeng Xu. Clustering-based cross-sectional regime identification for financial market forecasting. In *Database and Expert Systems Applications: 33rd International Conference, DEXA 2022, Vienna, Austria, August 22–24, 2022, Proceedings, Part II*, pages 3–16. Springer, 2022.

[28] Shun-ichi Amari, Frank Nielsen, and Richard Nock. On clustering histograms with k-means by using mixed α-divergences. *Entropy*, 16(6):3273–3301, 2014.

[29] Attila Egri, Illés Horváth, Ferenc Kovács, Roland Molontay, and Krisztián Varga. Cross-correlation based clustering and dimension reduction of multivariate time series. In *2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES)*, pages 000241–000246. IEEE, 2017.

[30] Ángel López-Oriona and José A Vilar. Quantile cross-spectral density: A novel and effective tool for clustering multivariate time series. *Expert Systems with Applications*, 185:115677, 2021.

[31] Mauro Gallegati, Fabrizio Lillo, Rosario N Mantegna, and Vincenzo Tola. Cluster analysis for portfolio optimization. *Journal of Economic Dynamics and Control*, 32(1):235–258, 2008.

[32] Ernest P Chan. *Quantitative trading: how to build your own algorithmic trading business*. John Wiley & Sons, 2021.

[33] Mario Carrasco Blázquez, Camilo Prado Román, et al. Pairs trading techniques: An empirical contrast. *European Research on Management and Business Economics*, 24(3):160–167, 2018.

[34] Chulwoo Han, Zhaodong He, and Alenson Jun Wei Toh. Pairs trading via unsupervised learning. *European Journal of Operational Research*, 307(2):929–947, 2023.

[35] Jeremiah Green, John RM Hand, and X Frank Zhang. The characteristics that provide independent information about average us monthly stock returns. *The Review of Financial Studies*, 30(12):4389–4436, 2017.

[36] Paul SP Cowpertwait and Andrew V Metcalfe. *Introductory time series with R*. Springer Science & Business Media, 2009.

[37] Stephen Haben, William Holderbaum, and Marcus Voss. *Core Concepts and Methods in Load Forecasting: With Applications in Distribution Networks*. Springer Nature, 2023.

[38] Jos Alfredo F Costa, Jorge D de Melo, Ade M Martins, and Adriao DD Neto. Clustering using neural networks and kullback-leibler divergency. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 4, pages 2813–2817. IEEE, 2004.

[39] Rogier Brussee and Christian Wartena. Topic detection by clustering keywords. In *2008 19th international workshop on database and expert systems applications*, pages 54–58. IEEE, 2008.

[40] Giridharan Iyengar and Andrew B Lippman. Video and image clustering using relative entropy. In *Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 436–445. SPIE, 1998.

[41] Qiong Deng, Danyang Huang, Bingyi Jing, Bo Zhang, and Yingqiu Zhu. Clustering based on kolmogorov–smirnov statistic with application to bank card transaction data. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 70(3):558–578, 2021.

[42] David M Mason and John H Schuenemeyer. A modified kolmogorov-smirnov test sensitive to tail alternatives. *The annals of Statistics*, pages 933–946, 1983.

[43] Sylvain Arlot, Alain Celisse, and Zaid Harchaoui. A kernel multiple change-point algorithm via model selection. *Journal of machine learning research*, 20(162), 2019.

[44] Ali Ghodsi, Karsten Keller, and Mathieu Sinn. Detecting change-points in time series by maximum mean discrepancy of ordinal pattern distributions. *arXiv preprint arXiv:1210.4903*, 2012.

[45] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Taylor & Francis online*, 1973.

[46] Preeti Arora, Shipra Varshney, et al. Analysis of k-means and k-medoids algorithm for big data. *Procedia Computer Science*, 78:507–512, 2016.

[47] David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 144–153, 2006.

[48] IBM. What is unsupervised learning? 2023.

[49] Patric Bonnier, Patrick Kidger, Terry Lyons, Imanol Perez Arribas, and Cristopher Salvi. Deep signature transforms. *Advances in Neural Information Processing Systems*, 32, 2019.

[50] David Arthur and Sergei Vassilvitskii. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

[51] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[52] Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Bharath K Sriperumbudur. Characteristic kernels on groups and semigroups. *Advances in neural information processing systems*, 21, 2008.

[53] Leonid V Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.

[54] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704, 1781.

[55] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

[56] Matthew Thorpe. Introduction to optimal transport. *Lecture Notes*, 3, 2019.

[57] Luigi Ambrosio, Alberto Bressan, Dirk Helbing, Axel Klar, and Enrique Zuazua. *Modelling and Optimisation of Flows on Networks: Cetraro, Italy 2009, Editors: Benedetto Piccoli, Michel Rascle*, volume 2062. Springer, 2012.

[58] Roland Badeau, Soheil Kolouri, Kimia Nadjahi, Gustavo Rohde, and Umut Simsekli. Generalized sliced wasserstein distances. *Advances in neural information processing systems*, 32, 2019.

[59] Yoshua Bengio, Aaron Courville, and Ian Goodfellow. *Deep learning*. MIT press, 2016.

[60] Gérard Letac and Paul Malliavin. *Integration and probability*, volume 157. Springer Science & Business Media, 1995.

[61] Gerald B Folland. *Real analysis: modern techniques and their applications*, volume 40. John Wiley & Sons, 1999.

[62] Daniel Revuz and Marc Yor. *Continuous martingales and Brownian motion*, volume 293. Springer Science & Business Media, 2013.

[63] Marc Bernot, Julie Delon, Gabriel Peyré, and Julien Rabin. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*, pages 435–446. Springer, 2012.

[64] Erhan Bayraktar and Gaoyue Guo. Strong equivalence between metrics of wasserstein type. *Project Euclid*, 2021.

[65] Nicolas Bonnotte. *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Université Paris Sud-Paris XI; Scuola normale superiore (Pise, Italie), 2013.

[66] Marco Cuturi and François-Pierre Paty. Subspace robust wasserstein distances. In *International conference on machine learning*, pages 5072–5081. PMLR, 2019.

[67] Marco Cuturi, Alexandre Gramfort, and Hicham Janati. Debiased sinkhorn barycenters. In *International Conference on Machine Learning*, pages 4692–4701. PMLR, 2020.

[68] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014.

[69] Jia Li, James Z Wang, Panruo Wu, and Jianbo Ye. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017.

[70] Dongdong Ge, Haoyue Wang, Zikai Xiong, and Yinyu Ye. Interior-point methods strike back: Solving the wasserstein barycenter problem. *Advances in Neural Information Processing Systems*, 32, 2019.

[71] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[72] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment problems: revised reprint*. SIAM, 2012.

[73] U Lall, As Sharma, and David G Tarboton. Kernel bandwidth selection for a first order nonparametric streamflow simulation model. *Stochastic Hydrology and Hydraulics*, 12:33–52, 1998.

[74] Rachel Campbell, Kees Koedijk, and Paul Kofman. Increased correlation in bear markets. *Financial Analysts Journal*, 58(1):87–94, 2002.

[75] Gerald HL Cheang and Carl Chiarella. Exchange options under jump-diffusion dynamics. *Applied Mathematical Finance*, 18(3):245–276, 2011.

[76] M Sklar. Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*, volume 8, pages 229–231, 1959.

[77] Yadolah Dodge. *The concise encyclopedia of statistics*. Springer Science & Business Media, 2008.

[78] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[79] Angel De la Fuente. *Mathematical methods and models for economists*. Cambridge University Press, 2000.

[80] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892, 2002.

[81] Bernhard Schölkopf and Alex Smola. Support vector machines and kernel algorithms. In *Encyclopedia of Biostatistics*, pages 5328–5335. Wiley, 2005.

[82] Rui Gao, Jie Wang, and Yao Xie. Two-sample test using projected wasserstein distance. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 3320–3325. IEEE, 2021.