ABSTRACT

PERRYMAN, BENJAMIN ALAN. Subsequence-Based Time Series Clustering Utilizing Stochastic Selection Methods. (Under the direction of Dr. Peter Bloomfield).

Time series clustering has become a topic of great interest in the data mining community. The increased need across a diverse set of fields has produced a plethora of different algorithms and approaches to producing information from temporal data sets. Recently, subsequence-based time series clustering has been developed to address complications which arise when a time series has repeated patterns, or motifs, punctuated by periods of relative noise. The recent rise in data streaming monitoring devices has further increased the need for time series classification using subsequences rather than full time series, due to limitations in storage capacity.

Subsequence approaches historically have had two focuses: motif discovery and motif membership evaluation. Using a single univariate time series for analysis, these approaches identify common structures within time series, or determine the presence of previously-known structures, respectively. The extension of this method to cases of multivariate data has been considered but is commonly handled by application of the univariate approach to each of the univariate time series of which the multivariate time series is composed.

This dissertation creates an end-to-end approach to subsequence-based time series clustering. The historically single stage process of subsequence clustering is extended to the identification of common time series, utilizing motif occurrence as a feature space on which to use static data clustering methods. The creation of stochastic processes to define motifs provides an increased understanding of the structures within these time series, and a low cost goodness of fit similarity measure is used to decrease the cost associated with membership

evaluation when approaching incremental data loads or data streaming. An extension of the univariate approach is presented which analyzes the multivariate time series data as a whole rather than considering each univariate time series separately. Case studies and test sets are used to demonstrate the effectiveness of this approach over a wide range of applications.

Subsequence-Based Time Series Clustering Utilizing Stochastic Selection Methods

by
Benjamin Alan Perryman

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Operations Research

Raleigh, North Carolina

2015

APPROVED BY:

_____	_____
Dr. Peter Bloomfield Committee Chair	Dr. Robert Abt
_____	_____
Dr. Anya McGuirk	Dr. Stephen Schecter
_____	
Dr. Ralph Smith	

DEDICATION

To keeping a promise to my five year old self.

BIOGRAPHY

Benjamin Perryman has always been a curious person. From an early age, he kept an invention journal, figuring out ways to make everyday items better. A deep interest in science and mathematics drove him to further his pursuits at the North Carolina School of Science and Mathematics for high school. He has since received a B.S. in Applied Mathematics from North Carolina State University, and a M.S. in Mathematics from the University of North Carolina at Chapel Hill. Ben is professionally interested in analytics, forecasting, and optimization. He enjoys billiards, rock climbing, martial arts, whiskey, and the conversations which result from them. Benjamin Perryman currently works as an analytical consultant at SAS Institute and resides in Raleigh, NC.

ACKNOWLEDGMENTS

A dissertation is not resultant of the efforts of one person alone. I would like to thank Dominick's grocery, and James M. Kilts Center, University of Chicago Booth School of Business for providing openly available sales data for research. The same gratitude goes to Duquesne Energy and North Carolina State University's Master of Analytics program for their energy load data. I hope more companies take their approach to providing information to guide or validate academic research in the future.

I would like to thank those who provided support and guidance through this process. Thank you Peter, Anya, Bob, Ralph, and Steve for being on my committee and providing me the guidance and assistance necessary to complete this work. Thanks to Dave Wear as well as the U.S. Forest Service for allowing me the work flexibility to begin this pursuit, and SAS Institute for allowing me to continue it. Thank you to all of the pool players who took time out of their Saturday to let me analyze their gameplay (See Table E.1). Thanks to my friends for pulling me away from my computer every now and again to keep me sane. A final thank you to my family for all of their love and patience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Introduction and Background

Clustering has become a topic in the forefront of data mining and data analysis. Clustering/classification has touched nearly every aspect of research, from analyzing precancerous lesions (Acosta-Mesa et al., 2014), to music classification (Fu et al., 2011b; Goulart, Guido, and Maciel, 2012), to early prediction of failure in wind turbines (Hoell and Omenzetter 2015; Qiu et al., 2012). Identifying structure in unlabeled data, clustering algorithms are utilized for attribute creation in hierarchical modeling, early warning detection, text analysis, and countless other applications (Chaovalitwongse et al., 2003; Liao, 2005).

Static data clustering methods have been investigated in depth over the course of the past half century. Partitioning clustering algorithms such as k-means which were developed nearly 40 years ago are now implemented into standard software packages with great usage in the scientific and business community (Kabacoff, 2014; SAS, 2015). Density based and hierarchical clustering practices have also become standard fare in the approach towards static clustering (Hennig, 2014).

The topic of time series analysis has become a focus of research in the field of clustering. The initial approach to time series classification was based upon the premise of constant signal, i.e. that all data points in a time series had equal importance. Euclidean distance measures were used to determine similarity, treating a time series sequence as a multidimensional data point, with comparison only possible given two sequences of equal length (Liao, 2005). During the 1990s and 2000s, clustering techniques for the purpose of

determining similarity between time series were created to generalize this approach for common data concerns (Moller-Levet et al., 2005). Dynamic Time Warping was created to address speed varying signals (Chan, Fu, and Yu, 2003; Jeong, S., M. Jeong, and Omitaomu, 2011). Strength of signal concerns lead to construction of shapelet and wavelet based algorithms (Chan et al., 2003; Ye and Keogh, 2011). Alternative similarity measures, using cross-correlation and goodness of fit, were created (Kumar and Patel, 2007; Liao, 2005). A comprehensive survey of previous approaches is given by Tak-chung Fu (Fu, 2011a).

During the course of the development of these time series clustering approaches, a secondary set of approaches was created based upon the alternative assumption that not all data in a time series was of equal worth. In some data cases, periods of relative noise are interrupted with signal, an indication of actors, or sub-processes, acting upon the variable of interest. This inconsistent signal can be seen in many natural processes such as punctuated equilibrium (Gould, 2007). Indeed, even in our daily lives, times of relative banality are broken with points of great interest. This new assumption of time series data structure produced the sub-field of subsequence-based time series clustering.

As a new field, the majority of research in subsequence clustering has been conducted within the past 10 years (Zolavarieh. Aghabozorgi, and Teh, 2014). Rather than focusing on similarity between time series, the previous goal, similarities between subsequences within a single time series are examined. Subsequence clustering allowed for analysis on data sets which were fractured, or incomplete, that would require preprocessing/imputation by the user in previous full time series approaches (Kabacoff, 2014). These subsequence approaches also allow for noncyclical behaviors to be cataloged, and are used for a fuller understanding

of actors on a system. An example of this, which is examined further in Chapter 5, is the effect of mixed marketing on sales of retail goods.

Similar to the explosion of different techniques which were created for full time series analysis, many methods were created for evaluation of similarity between subsequences and the determination of common patterns within time series, known as motifs. Approaches denoted bag-of-words (BOW), or bag-of-features, were created to address the time series data case in which the motifs were known or assumed, in which only classification of a subsequence as being similar to a motif was required to complete the analysis (Baydogan et al., 2013; Fu et al., 2011b). New similarity functions were created to assist with classification, such as radial distribution functions (Denton, Besemann, and Dorr, 2009). BOW classification strategies were expanded past character and numeric based evaluation, to image classification (Nowak, Jurie, and Triggs, 2006). These methods were useful in situations where motifs were previously known, such as word recognition for text analytics, but lacked the ability to discover new and unexpected motifs.

Motif discovery based algorithms became a popular topic of research to address cases of time series data sets with low prior knowledge. Generation of motifs for use in a bag-of-words approach produced a variety of algorithms (Abdulla_Al_Maruf, and Hung-Hsuan, 2012; Chen, Bi, and Wang 2006; Jadhav et al., 2013; Morrill 1998). A new similarity distance measure emerged, based on the amount of memory required to encode a time series. This memory, or description length, could be reduced by classifying multiple subsequences as having a single pattern. The memory to encode the pattern and the differences between the pattern and the true value of the subsequences can be less than the memory required to

encode each subsequence separately. The objective of maximal reduction in overall description length of a time series is the driver for the minimum description length (MDL) approach to subsequence clustering. A greedy iterative approach using this MDL objective was recently created to merge similar subsequences into new motifs, and add membership to pre-existing motifs (Rakthanmanon et al., 2012). This MDL principle has begun to be used in the subsequence clustering community (Mueen et al., 2010; Stine, 2004; Zolhavarieh, Aghabozorgi, and Teh, 2014). A comprehensive discussion of subsequence time series clustering is discussed in Zolhavarieh's review (Zolhavarieh, Aghabozorgi, and Teh, 2014).

Subsequence based methods have inherent hardships. High memory/computation is necessary to consider all available subsequences in motif discovery algorithms. Noisy data sources can have repeated patterns as a result of chance rather than true signal, leading to spurious results. Restrictive motif structure assumptions can prevent true subsequence signal from being identified in some cases, causing some to find the results of these subsequence clustering approaches meaningless (Keogh, and Lin, 2005; Zolhavarieh, Aghabozorgi, and Teh, 2014).

For all these short-comings, subsequence analysis is still progressing with great interest. The internet of things is fast becoming the new benchmark on which data mining algorithms are tested (Zheng et al., 2011). Device monitoring data comes in with such a high rate that memory storage for entire time series is no longer possible (Sanniano, De Falco, and De Pietro 2014; Silva et al., 2013). Subsequence motifs allow for partial data sets to be kept, with summary tables of occurrences of motifs for time series to be tabulated, reducing computational cost. New similarity searches and definitions of motifs have been created to

reduce the complexity of computation associated with subsequence analysis (Lian, and Chen, 2008; Nguyen, Ng, and Woon, 2014). As a result of this market need, many packages have been created to address the issue of time series data stream clustering (Pereira and De Mello, 2014; Silva et al., 2013).

Multiple data sources can add to the complexity of data stream analysis. The issue of multivariate data is not a new one. In raw data time series analysis, cross-sectional approaches were used to address this issue (Kosmelj and Batagelj,1990). In subsequence clustering, univariate studies continue to be the focus, analyzing multivariate time series using univariate clustering techniques on each variable separately. The resultant multivariate time series analysis is created from the compilation, or "stacking", of these univariate results (Alonso et al. 2008; Liao 2005). This stacking lacks allowance for interplay between variables in a multivariate time series analysis, resulting in an incomplete understanding of the underlying sub-processes.

1.2 Overview of Dissertation

This dissertation presents and demonstrates a subsequence-based approach to time series clustering, allowing for the flexibility of motif creation with the computational efficiency of bag-of-words subsequence membership identification, utilizing goodness of fit measures with stochastically defined motifs. The relative occurrence rate of motifs for each time series serves as a feature space, allowing for static clustering methods such as hard k-means, fuzzy c-means, and Gath-Geva to be utilized to group similar time series. Incremental load approaches are presented, utilizing the low cost goodness of fit calculation

to create quick subsequence membership updates, satisfying the growing needs of time series data streams.

The presented approach initially utilizes motif discovery on full time series data through an MDL-based greedy iterative approach, similar to that created by Rakthanmanon et al. (2012). Improvement on the subsequence cluster merging methodology removes redundant clusters. Additionally, this modification allows for subsequence clusters with partially representative patterns to transfer membership to the full pattern subsequence clusters. Business user defined motifs are added prior to the MDL approach, allowing for hypothesized motifs to be included in the MDL process, reducing the considerable computational time associated with motif discovery.

Completion of the motif discovery phase initializes the creation of stochastic model representations of the subsequence patterns. Multiple approaches to stochastic model creation are discussed in Chapters 2 and 4, to cover some of the diverse data cases which can occur. After stochastic process estimation, candidate subsequences are able to be evaluated based upon chi-squared or F-like tests, adding/removing membership with much greater speed than allowable via the MDL-based evaluation method of bitsave. The approach is discussed in detail and tested in Chapter 2, with multiple applications demonstrating the effectiveness of this procedure given in Chapter 3.

In Chapter 4, the univariate approach of Chapter 2 is extended to a multivariate approach. Rather than stacking univariate results as done often in the previous subsequence-based clustering literature, the notion of a motif is extended into multiple dimensions. Two frameworks for a truly multivariate algorithm are presented, hinging on differing base

definitions of the structure of a multivariate motif. A hybrid approach between these two methods is reached, allowing for a linear increase in computational complexity relative to the number of variables to be reached in motif discovery, and a low-cost, higher accuracy definition of motif to be utilized for incremental loads and/or data streams. This approach has lower computational complexity associated with stacking univariate results, and allows for the interplay of multiple dimensions to define a motif. Real world multivariate applications are discussed in Chapter 5.

1.3 Extensions to Current Field

The approach outlined in this dissertation extends the current research in a number of ways. Previous work on subsequence clustering has been focused on determining underlying patterns within a single time series, with little research into usage of the relative occurrence of these motifs as a feature space to cluster similar time series together. Additionally, defining motifs by stochastic processes allow for a greater comprehension of the underlying processes and assists in the production of low-cost similarity measures such as chi-square goodness if fit and F-like tests which reduce computation time and allow user-defined similarity thresholds for membership evaluation.

The approach discussed in Chapter 4 expands upon the notion of motifs to include multivariate motif definitions, a concept not extensively researched in the subsequence-based time series clustering literature. Generalization of the definition of a motif to include nonconsecutive values allows for a fuller understanding of the sub-processes acting upon a time series, also creating an approach which increases the level of knowledge gained from a multivariate time series stream.

CHAPTER 2: UNIVARIATE SUBSEQUENCE-BASED TIME SERIES CLUSTERING

2.1 Introduction

Time series analysis is a topic of great interest across industries. An unprecedented level of time series information is being gathered and analyzed, from process information in manufacturing plants, to point of sales data in retail. In an effort to gain knowledge from these temporal data sets, much work has been put into machine learning and clustering (Liao 2005). As discussed in Chapter 1, two families of approaches to time series clustering have emerged: the first clustering using the entire time series for analysis (Fourier Transforms, Wavelets, etc.), and the second clustering occurrences of common subsequences within a series (Bag-of-Words, MDL-based motif discovery) (Liao, 2005; Lin et al., 2007).

Continual monitoring systems have recently become a topic of great interest, in which time stamped data streams produce time series too large for full storage (Zheng et al., 2011). This business need to not retain full time series information has created a heightened interest in subsequence-based time series clustering. This chapter outlines a comprehensive univariate subsequence-based approach to time series clustering.

Using the approach outlined in this chapter, similar time series will be clustered based on the relative occurrence of common subsequence patterns, known as motifs. In order to properly cluster time series in this manner, two issues need to be resolved. The first issue to be resolved is an understanding of which motifs exist within the time series data. The second issue for resolution is determining which subsequences within a time series are sufficiently similar to these motifs.

The first issue of subsequence clustering can be addressed with one of two approaches: user defined motifs, or motif creation through discovery (Baydogan et al., 2013; Rakthanmanon et al., 2012). User-defined motifs can be useful, but suffer from the potential for an incomplete set of motifs, as well as the possibility of those motifs being of poor/biased quality. Alternatively, motif discovery creates subsequence clusters from similar subsequence occurring in the data. The associated motif to this subsequence cluster is equal to the average of the subsequences with membership to the cluster, producing an unbiased pattern occurring in the time series.

The second issue of subsequence clustering is addressed through similarity measures. Euclidean distance, dynamic time warping distance, and description length are just a few of the measures which have been utilized to determine the similarity between two subsequences, or a subsequence and a motif (Zolhavarieh, Aghabozorgi, and Teh, 2014). One approach to motif discovery uses description length as the basis of a greedy iterative algorithm which minimizes the overall description length of a time series (MDL) (Rakthanmanon et al., 2012).

Using the natural concept of reduction in description length associated with any creation or update to a subsequence cluster, this iterative algorithm can provide a comprehensive list of motifs occurring in the data set. A user-defined motif input prior to a MDL-based algorithm will be the basis of the motif creation/discovery portion of the approach outlined in this chapter.

Upon completion of motif discovery, the subsequence members of a cluster can be considered realizations of a stochastic process. Based upon the data type and appropriateness

of assumptions, a choice of type of stochastic process is made by the user. Estimation of the parameters associated with the stochastic process is performed to create a hypothesized distribution of the subsequence cluster pattern. This constructed distribution allows for chi-square goodness of fit or F-like tests to be used on eligible subsequences to provide a quick and informative measure of similarity. Given subsequences which are not significantly dissimilar from the hypothesized distribution, using a user-defined threshold for level of significance, these subsequences can be added to the most representative subsequence cluster.

The completion of motif discovery, stochastic process definition/estimation, and membership additions/removals using the goodness of fit similarity measure, allows for the creation of a motif occurrence rates for each time series. Using these motif occurrences as a feature space, a choice of static clustering approaches is selected to group similar time series, completing the full approach.

Section 2.2 discusses all background information necessary for creation of the subsequence-based time series clustering approach. Section 2.3 explains the methodology of this approach in detail, explaining the process in 5 distinct phases. Additionally, this section discusses incremental updates, in which the value of having the low-cost goodness of fit similarity measure is truly realized. The complexity of each of the phases from Section 2.3 is discussed in Section 2.4.

To demonstrate the effectiveness of this approach, test data time series are created using motifs embedded in noise. Using Case 1 of Phase 3, discussed in detail in Section 2.3, clustering occurs using the fully operational code built in SAS® base language. An

explanation of the construction of the datasets used for these tests is given in Section 2.5 and the associated clustering results are given in Section 2.6.

2.2 Background and Notation

2.2.1 Minimum Description Length Process

Definition 2.1- Time Series/Length

A *time series* T is a sequence of values $t_1, t_2, \dots, t_n$ representing the realization of a variable at each of the n points in time. The *length* of the time series T is defined as n.

Definition 2.2- Subsequence

A *subsequence* $T_{i,k} = t_i, t_{i+1}, \dots, t_{i+k-1}$ is a group of values within the ordered sequence T, of length $k \leq n$ which is ordered as in T without exclusion. This definition requires $i \in \{1,2, \dots, n-k+1\}$.

A time series is not explicitly required to have an evenly spaced set of observations. However, for the purposes of our study we assume even spacing. Additionally, continuous valued time series data is required to be discretized for the purposes of the MDL algorithm, with values between 1 and 2^b. The constant b, defined as the *level of resolution*, allows for a lossy discretization/normalization of the subsequence data. The purpose of this discretization/normalization is two-fold. Firstly, the normalization allows for the comparison of two subsequences based on their relative shapes, removing issues of scale. Secondly, the discretization allows for a computation of the number of bits required to encode the difference between a subsequence and a motif. The constant b represents the maximum number of bits used to encode the lossy approximation of a single data point in the time series. The discretization/normalization is defined as follows:

Definition 2.3- Discrete Normalization

Given a continuous valued subsequence T, the discretized sequence representation is produced by the *discrete normalization* function DNorm, defined as:

$$DNorm(T) = round\left(\left(\frac{T - Tmin}{Tmax - Tmin}\right)x(2^b - 1)\right) + 1,$$

where Tmin and Tmax are the minimum and maximum values of the sequence T. The result of this transformation is that all values fall between 1 and 2^b. When Tmax=Tmin, DNorm(T) will be set to a vector of 2^{b-1}s.

The normalization of subsequences allows for similarity of shape to be assessed. One quick assessment of the similarity of curves for two subsequences of the same length is by the Euclidean norm. While not used for the final membership assessment in this chapter's approach, the Euclidean distance measure provides a low computational cost method to assess similarity, with results similar to that of using description length (Rakthanmanon et al. 2012).

Definition 2.4-Euclidean Distance

The Euclidean distance between subsequences $T_{i,k}$ and $S_{j,k}$ is:

$$EDist(T_{i,k}, S_{j,k}) = \sqrt{\sum_{l=0}^{l-1}(t_{i+l} - s_{j+l})^2}$$

The underpinning of the description length is the measurement of entropy. An intuitive definition of entropy is that it is the level of disorder in a system. In the case of a time series, this is observed as the level of noise (differentiation of values).

Definition 2.5-Entropy

The entropy of a time series T is defined as follows:

$$H(T) = -\sum_{t} P(T = t) \log_2 P(T = t),$$

where P(T=t) is the probability that a randomly chosen value in the series T has a value of t, and with the convention that if P(T=t*)=0, then $P(T = t*) \log_2 P(T = t*) = 0$.

If there are a set of subsequences which have a common pattern, then encoding the information of those subsequences by a motif and the associated deviation of the subsequences from that motif could take up less total memory than encoding each of the subsequences separately. This reduction in encoding memory is caused by a fewer number of bits being required to encode the associated differences rather than the original subsequences. The entropy can be interpreted as a measure of the amount of memory required to encode each position of a time series. The description length measures the relative amount of memory required to encode the entire series T.

Definition 2.6-Description Length

The *description length* of a time series T of length n is:

$$DL(T) = n * H(T)$$

Definition 2.7- Subsequence Cluster/Motif of Subsequence Cluster

A *subsequence cluster* C_H *of length n* is a set of subsequences $\{SS_{H,1}, SS_{H,2}, \ldots, SS_{H,k}\}$ each of length n. A *motif H associated with subsequence cluster* C_H is the average subsequence value of the members of the cluster.

Intuitively, a motif is a repeated pattern in the time series. Motifs need not be defined by a subsequence, but can be defined solely by the pattern, as is the case in bag-of-words approaches. In this case, the motif is not affected by the realizations of the associated subsequences, which allows for subsequences to deviate from the motif in a biased manner. These business-user-defined motifs are discussed further in Section 2.3.1.

Definition 2.8-Conditional Description length

The conditional description length of a subsequence B associated a motif H is:

$$DL(B|H) = DL(B - H)$$

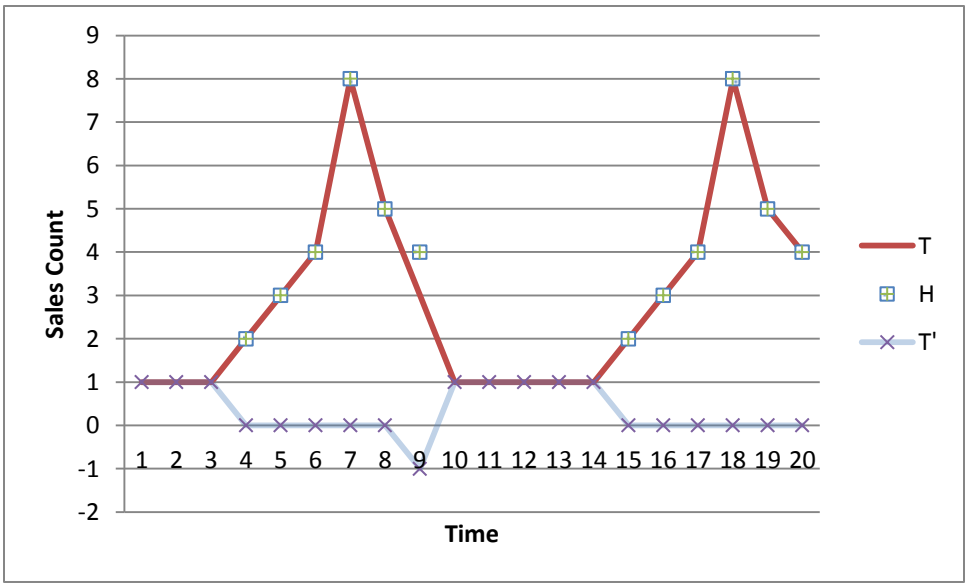


Figure 2.1 - T is partly represented by the motif H to produce T'

Using conditional description length definition, the usefulness of a motif becomes clearer. As an example, Figure 2.1 displays the time series T of the number of car sales over the course of 20 weeks. $T_{4,6}$ and $T_{15,6}$ display a near exactly recurrent subsequence pattern.

Creating a subsequence cluster C_H={$T_{4,6}$, $T_{15,6}$} results in the motif H=2, 3, 4, 8, 5, 4 (rounding to nearest integer).

The time series can be modified by the subsequence cluster C_H. T' is the resultant series, taking the difference of T and H for $T_{4,6}$ and $T_{15,6}$. For notation,

$$T' := (T|H) = \begin{cases} T - H, for\ starting\ points\ t = \{4,15\} \\ T, otherwise \end{cases}$$

Examining T' in Figure 2.1, there are fewer realized values in T'. As a result, it is possible to encode T' into memory using fewer bits, measured by a lower resultant description length. Calculating the description lengths:

DL(T)=20*.70635=14.127, DL(T')=DL(T|H)=20*.36703=7.341, DL(H)=6*.67781=4.067

The original description length for T is 14.127. Using subsequence cluster C_H to reduce the entropy of T, T' has a description length of only 7.341! There is the additional cost of the description length for H of 4.067, producing a total description length of encoding T using C_H of 11.408. The MDL approach seeks to minimize the total description length of a time series T by using subsequence clusters. As a result, MDL would determine the motif H defined by C_H to be a significant pattern within the series T, and C_H would be utilized.

Definition 2.9-Description Length of a Cluster (DLC)

The *description length of a cluster* C_H with centroid H and members $A \in C_H$ is:

$$DLC(C_H) = DL(H) + \sum_{A \in C} DL(A|H)$$

The description length of a time series can be altered by the use of subsequence clusters. It is necessary to quantify the usefulness of changes to the set of subsequence

clusters associated utilized by a time series T. Three actions can be performed on the set of subsequence clusters. These actions are forming a new subsequence cluster defined by two subsequences in T, adding a subsequence of T to a preexisting cluster, or merging the membership sets of two clusters together. Each of these actions alters the description length of T, and the extent of the change is defined by the bitsave.

Definition 2.10-Bitsave

The *bitsave* associated with an action on the set of subsequence clusters is the difference in the description length of T before versus after the action is used. Namely,

$$bitsave = DL(T|Clusters\ before\ action) - DL(T|Clusters\ after\ action)$$

An action with positive bitsave reduces the description length of T and is a candidate for use. The greedy approach to MDL chooses the action with the greatest bitsave, and if the associated bitsave is positive, uses this action upon the set of subsequence clusters. A detailed discussion of the actions available is given in Section 2.3.

This subsection gave the framework required for motif discovery and refinement. As discussed further in Section 2.3.4, the members of the subsequence clusters can be considered as realizations of a stochastic process. Based on the data type being analyzed, different stochastic processes may be more apt to represent the subsequence cluster. In Section 2.2.2, Poisson processes are discussed, which are used for counted data when certain conditions hold. Section 2.2.3 discusses Gaussian processes in particular cases of continuous data.

2.2.2 Poisson Processes

Definition 2.11-Poisson Distribution (X~Pois(λ))

A discrete random variable X has a Poisson distribution with occurrence rate λ if it takes on values of nonnegative integers k=0,1,2,… with probability function:

$$f(k,\lambda) = \Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Definition 2.12-Time Homogeneous Poisson Process

A Time Homogeneous Poisson process, N(t), is a stochastic counting process of the number of occurrences of an event within the time span (0,t], in which the time between occurrences, X, is exponentially distributed with parameter λ. The probability of k occurrences between times t and t+h follows a Poisson distribution with occurrence rate λh. Namely,

$$P[N(t+h) - N(t) = k] = \frac{\lambda h^k e^{-\lambda h}}{k!}$$

In order for a counting process N(t) to be a Time Homogeneous Poisson, only 4 assumptions need to be satisfied (Varahan, 2000):

i) For every h>0, the distribution of N(t+h)-N(t) is the same for every t>0 (the time homogeneity assumption).

ii) If (a,b] and (c,d] do not overlap, then the random variables N(b)-N(a) and N(d)-N(c) are mutually independent (Memory-less).

iii) N(0)=0, N(t) integer valued, right continuous, non-decreasing in t with probability equal to 1 (Properties of a counting process).

iv) P[N(t+h)-N(t)≥2]=P[N(h)≥2]=o(h) as h→0 (Sparse events).

For a counting process in many industry functions, (iii) is applicable if negative counts are not allowed or are negligible (e.g. returns of merchandise). Assumption (iv) is also often reasonable, given events which do not occur densely. Note that there can be occurrences in which multiple events occur at the same time, but if they are rare it is possible to model them as being a small distance apart, especially when looking at time aggregated data.

In some cases it may be that the counting process is dependent on time. In these cases assumption (i) is not satisfied and the result is a time nonhomogeneous Poisson process.

Definition 2.13-Time Nonhomogeneous Poisson Process (NHPP)

A Time Nonhomogeneous Poisson process (NHPP), N(t), is a Poisson process in which the occurrence rate λ now varies with time. Define the occurrence rate as $\lambda(t)$, and the occurrence rate over the interval t=(a,b] as:

$$\Lambda_{a,b} = \int_a^b \lambda(t)dt$$

The NHPP has the property that the likelihood of k occurrences in the interval t=(a,b] has a Poisson distribution with occurrence rate $\Lambda_{a,b}$:

$$P[N(t+h) - N(t) = k] = \frac{{\Lambda_{a,b}}^k e^{-\Lambda_{a,b}}}{k!}$$

2.2.3 Gaussian Processes

Definition 2.14-Gaussian Distribution ($X \sim N(\mu, \sigma^2)$) (Gallager 2013)

A continuous random variable X is *Gaussian* with mean μ and variance σ^2 if it satisfies the density function:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Definition 2.15-Gaussian Process (Gallager 2013)

A Gaussian process $\{X(t);t \epsilon T\}$ is a stochastic process such that for any set of k time epochs, $t_1, \ldots, t_k \epsilon T$, the set of random variables $X(t_1),\ldots, X(t_k)$ is a jointly-Gaussian set of random variables.

2.3 Methodology

The approach outlined in this section provides a comprehensive end-to-end approach to subsequence-based time series clustering. The approach is broken into five phases, each of which addresses a particular concern inherent in time series clustering. Phase 0 allows for the input of user-defined motifs with levels of confidence in the motif chosen. These user-defined motifs create artificial subsequence clusters. Phase 1, the first MDL-based motif discovery phase, acts upon the time series, creating new subsequence clusters in addition to the user defined motifs, and adding membership to preexisting subsequence clusters. Phase 2 Part 1 merges subsequence clusters together, creating new candidate subsequence clusters. At this point, the MDL-based approach laid out in Rakthanmanon's work is completed. Additional steps beyond Rakthanmanon's work are given in Phases 2 Part 2, 3, and 4.

Discussed in Section 2.3.3.2, the subsequence cluster merging in Phase 2 Part 1 can produce motifs not naturally occurring in the data. To combat these bad motifs, and to reduce the set of clusters to only those with distinct motifs, Phase 2 Part 2 reevaluates subsequence membership to clusters using a hierarchical subsequence cluster scheme. Phase

3 creates stochastic models for each subsequence cluster of sufficient size, using the membership subsequences as realizations of the process. These stochastic models create hypothesized distributions on which dissimilarity can be assessed, using chi-square goodness of fit or an F-like test statistic, for new eligible subsequences.

Phase 4 accumulates the relative subsequence membership rates for each time series being considered, and uses these rates as a feature space. Partition clustering is performed on this feature space to create groupings of similar time series. Pseudocode for this approach is given in Appendix A.

This algorithm is customizable to fit the needs of a particular data set. The parameters available for modification are:

1. The motif window [MMmin, MMmax].
2. Iteration cutoffs for Phases 1 and 2 Part 2 (used in the case of time restrictions).
3. The minimum subsequence cluster size to be considered for stochastic modeling in Phase 3.
4. The threshold for the dissimilarity required for a subsequence to not become a member of a subsequence cluster, used in the goodness of fit test (alpha level).
5. The maximum number of series clusters to be used in Phase 4.
6. The partition clustering algorithm to be used on the subsequence cluster occurrence feature space in Phase 4.

The motif window defines the minimum and maximum lengths available for motif creation. Repeated subsequence patterns with length outside of this range may be partially

discovered by subsequence clusters, realized through multiple distinct subsequence clusters as is demonstrated in Section 2.6.

2.3.1 Phase 0: Business User Input

Unsupervised machine learning has many attributes to its benefit. It is attractive to think about an algorithm which can be applied to an arbitrary set of time series data, information gathered without any prior knowledge or hypotheses, the results of this approach creating meaningful knowledge which drives innovation and discovery. There are two criticisms to the unsupervised learning approach. The first is that there may be false artifacts, patterns found in the noise, which appear to be significant. When chosen, these erroneous motifs can stifle the learning process, especially with an exclusionary greedy algorithm such as MDL. The second criticism to the automated approach is that a data set is rarely unknown. In many business cases, the variable of interest in a time series is accompanied with expert theories on the sub-processes acting upon the variable of interest and potential patterns which can occur. In Phase 0, these theories are used to create initial guesses at behaviors expected to be occurring in the data, through user-defined motifs. These user-defined motifs are assessed in Phases 1 and 2, adding subsequences to membership as appropriate. If there is sufficient subsequence membership at the end of Phase 2, a stochastic process representation of this motif is created in Phase 3, and the motif occurrence rate is used in Phase 4. Usage of user-defined motifs addresses some of the criticisms associated with automated motif discovery, and leads to increased speed in Phases 1 and 2.

User defined motifs are assigned a confidence level. This level is a representation of the accuracy associated with the motif pattern. This value acts as the size of the artificial

subsequence cluster with the associated user defined motif. Another way to describe this approach is that given a confidence level of x, x artificial subsequences equal to the motif are created and given membership to the cluster. A lower size (1 or 2) allows for a greater level of adjustment from the real data to account for inaccuracies than would be possible with a size of 1000. Indeed, due to the rounding and normalization occurring after each action in Phase 1, placing a sizing of greater than 2b-3 ensures that the motif does not change, where b is the level of resolution.

2.3.2 Phase 1-MDL Cluster Creation

Given the initialization of user-defined clusters (motifs along with associated confidence sizing), the modified MDL algorithm begins. As stated in Section 2.2, the concept underlying MDL is minimizing the amount of memory (description length) required to describe the time series. Description length is reduced iteratively, choosing an action with the greatest positive bitsave to use upon the set of subsequence clusters. In Phase 1, two actions are possible for use, a modification of the approach by Rakthanmanon et al. (2012).

Action 1: The merging of two subsequences-

> Two subsequences of the same length create the membership set of a subsequence cluster with motif (centroid) equal to the average of these two subsequences. There is an associated subsequence cluster size of 2 (each subsequence unto itself is considered to have a size of 1 and size is additive).

Action 2: A subsequence is added to the membership set of a subsequence cluster C_H.

If C_H is previously of size x, length n, and motif $M=m_1, m_{2, \ldots,} m_n$, the addition of subsequence $T_{i,n}=t_{i,1}, t_{i,2},\ldots,t_{i,n}$, updates C_H to a cluster of size x+1, with centroid $M'=(xM+ T_{i,n})/(x+1)$.

For a subsequence to be considered for these actions, no portion can have previously been assigned membership (no overlapping subsequences). A subsequence which does not have any data points with membership to subsequence clusters is denoted *actionable,* or *eligible*. At each iteration of the MDL algorithm, the two nearest actionable subsequences are found using the low-cost Euclidean distance metric, for each length L in the motif window. The bitsave associated with Action 1 on these subsequences is calculated and stored for review. For each cluster, the nearest actionable subsequence is also selected using the Euclidean distance. The bitsave associated with Action 2 on this subsequence and cluster is assessed and stored for review.

Upon completion of all bitsave calculations, the action with the largest bitsave is chosen. If the bitsave is positive (a reduction in the overall description length of the data), then the associated action is completed. Eligibility of actionable subsequences is reassessed after the update to the set of subsequence clusters, and the next iteration begins. These iterations continue until the greatest bitsave is negative, or the maximum number of iterations possible is reached. The setting of maximum number of iterations is solely for the purpose of time consideration.

Upon completion of the iterative process, a set of candidate clusters with associated motifs have been created, size indicating the prevalence of these subsequence patterns in the time series data. Phase 1 does not exclude the ability for two subsequence clusters to have

similar motifs, but does give preference to Action 2 over Action 1, given the same bitsave. In order to compress similar clusters, Phase 2 allows for usage of the third action: merging membership sets of similar clusters.

2.3.3 Phase 2-Cluster Pruning

2.3.3.1 Phase 2 Part 1-Merging similar clusters

Phase 1 produces a set of subsequence clusters, some of which may lack distinctness, or overlap significantly with another cluster within the set. To remedy these issues, a third action is created: merging two clusters together. This action is defined in a similar bitsave framework to the first two actions.

Action 3: Merging of two clusters-

Given two subsequence clusters C_1 and C_2, with lengths L_i, sizes S_i, motifs M_{i,L_i}, and an offset O_i, the resultant cluster C_3 has size S_1+S_2, length $L_3=\max(L_1+O_1,L_2+O_2)$, and motif:

$$m_{3,j} = \begin{cases} \dfrac{S_1 m_{1,j-O_1} + S_2 m_{2,j-O_2}}{S_1+S_2}, if\ O_1 < j, O_2 < j, (j-O_i) \le L_i \\ m_{1,j-O_1}, if\ O_1 < j, O_2 \ge j, (j-O_1) \le L_1 \\ m_{1,j-O_1}, if\ (j-O_1) \le L_1, O_2 + L_2 < j \\ m_{2,j-O_2}, if\ O_2 < j, O_1 \ge j, (j-O_2) \le L_2 \\ m_{2,j-O_2}, if\ (j-O_2) \le L_2, O_1 + L_1 < j \end{cases},$$

where the offset O_i is a nonnegative integer indicating number of positions to the right of the first position of M_3 that M_i is shifted. An intuitive explanation of Action 3 is that it is the combination of two clusters such that the motifs are overlaid with some offset, and the new motif is the size weighted average of the previous motifs. An example of the overlay process is given in Figures 2.2 A-C.

Given the constraints on motif length, the overlaps O_i are restricted such that L_3 is within the motif window. In order to assess the bitsave associated with this merger, the difference of each subsequence associated with either motif cluster is calculated, based on the relevant portion of M_3. The description length prior to Action 3 is $DLC(C_1) + DLC(C_2)$. The new description length $DLC(C_3)$ is calculated under the caveat that the conditional description lengths of the subsequence members use only the applicable portion of M_3. The bitsave of merging two similar clusters together is positive when there is high similarity of members along that overlap. In these cases, the increased conditional description costs associated with changing the motif value along the overlay is offset by the reduced description length of motif M_3 as compared to that of M_1 and M_2.

There is the potential for erroneous clusters to be created using Action 3. The bitsave calculation described above, based on the approach by Rakthanmanon et al., does not check to determine if the entire motif M_3 occurs in the dataset. Indeed, it is possible that M_3 can be produced from a partial overlap of M_1 and M_2, with no difference in overlaid values, and M_3 never occurs within the time series data.

An example of this phenomenon is shown in Figures 2.3 A-D. In Figure 2.3 clusters C_1 and C_2 each have size 2 and have perfect membership (member's difference from motif is 0). When O_1=0, O_2=3, the overlap of M_1 and M_2 matches perfectly. M_3 maintains the zero difference description length for all subsequences associated with C_1 and C_2, while reducing the description length of motif encoding. Action 3 on these clusters will result in positive calculated bitsave! As a result Action 3 will occur and C_3 will be produced as a candidate cluster, but M_3 does not show up anywhere in T, shown in Figure 2.3A.

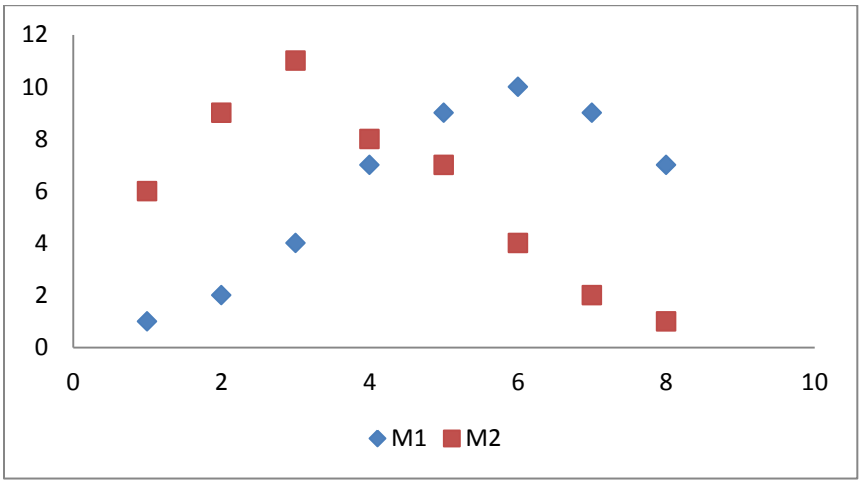


Figure 2.2A - Two motifs from Phase 1. S_1=5, S_2=8, O_1=0, O_2=0

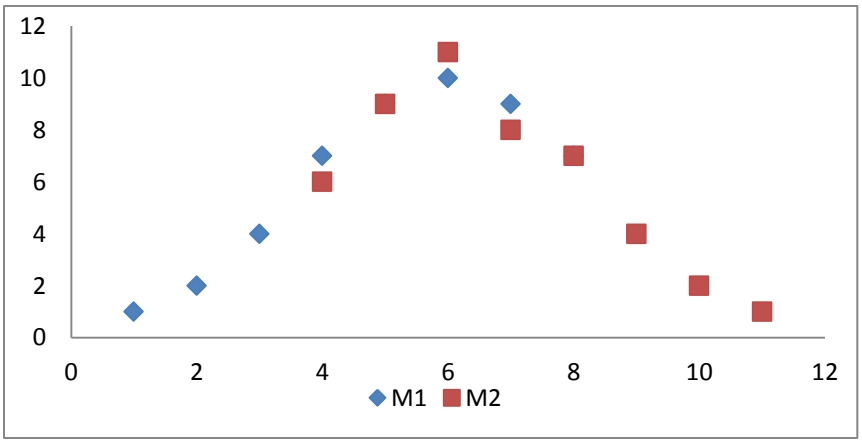


Figure 2.2B - Offset second cluster. O_2=3

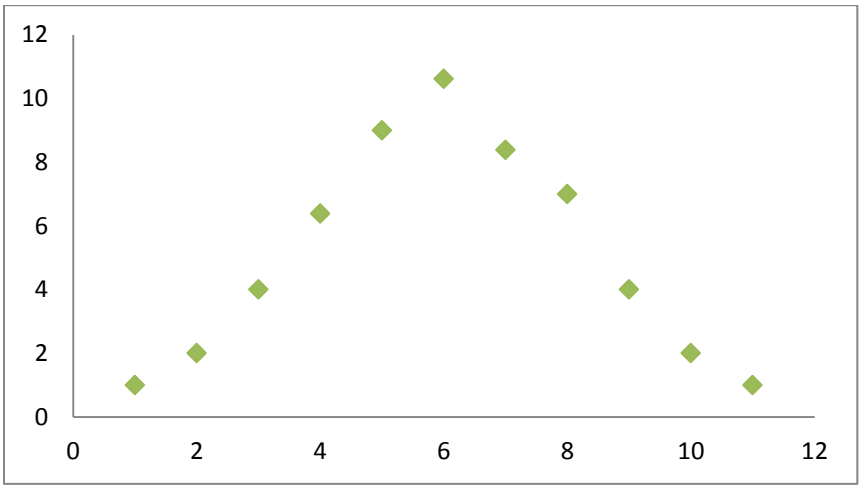


Figure 2.2C - M_3 created from M_1 and M_2 with O_1=0, O_2=3

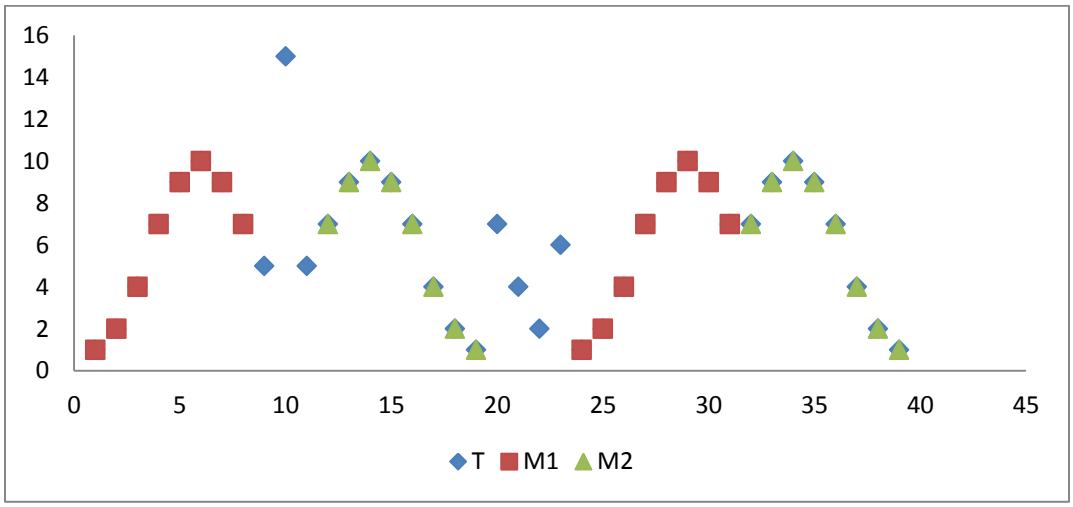


Figure 2.3A - Time series T with associated motif cluster A and B membership shown

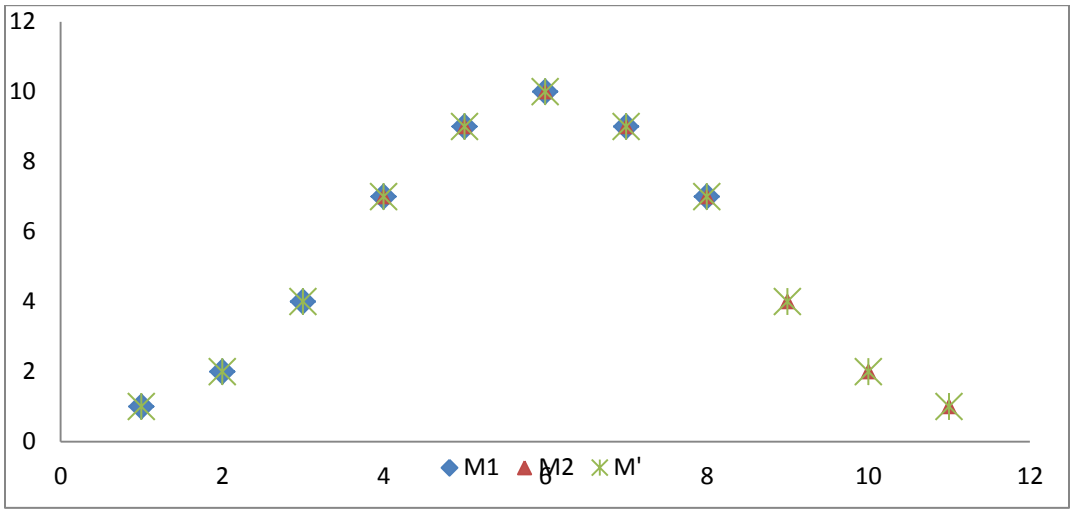


Figure 2.3B - M_3 made from Highest Bitsave associated Action 3 (O_1=0, O_2=3)

Unfortunately, Action 3 does not always produce a cluster with a motif that actually occurs in the time series data. It does always produce a candidate cluster which has a length larger than or equal to each of the initial clusters and a good motif representation of member subsequences on the overlap. As a result, if subsequences do occur naturally with this motif, representing those subsequences by this amalgamated cluster can reduce description length. Phase 2 Part 2 was created to determine which of these new motifs occur in the data.

In order to determine all candidate clusters which may be represented in the data, an iterative approach to merging clusters is taken in Phase 2 Part 1. Beginning with all clusters from Phase 1 being set to active, the maximum bitsave of action 3 of each merger is assessed (varying on the offsets). The largest bitsave associated with Action 3 across all pairings is determined, and if positive, that instance of Action 3 is taken. Without loss of generality, let C_1 and C_2 be the clusters on which Action 3 is carried out, with the resultant cluster denoted C_3. C_3 is set to active, with cluster size equal to the sum of the sizes of C_1 and C_2. The

merger size of C_3, denoted MS_3, is defined to be the number of clusters from Phase 1 which were merged to create C_3. Therefore $MS_3 = MS_1 + MS_2$, using the convention that any subsequence clusters from Phase 1 have $MS_i = 1$. C_1 and C_2 are then set to inactive to complete the iteration. This process iterates on active clusters until there is not a positive bitsave associated with any instances of Action 3, or there is only one active cluster. The output of this phase is a list of all subsequence clusters.

2.3.3.2 Phase 2 Part 2-Reselecting Subsequence Membership

Phases 1 and 2 Part 1 produced a plethora of candidate subsequence clusters. Phase 2 Part 2 reduces the overall number of subsequence clusters, determining which distinct clusters are represented in the time series data. Facilitating this reduction of clusters requires subsequence membership be reassessed. For each subsequence cluster C_H resultant from Phases 0 to 2 Part 1, the associated motif H is used to create an artificial subsequence cluster $C_{H,A,}$ with a confidence level/cluster size of 2b-3. Let the merger size transfer from C_H to $C_{H,A}$. Membership to all subsequence clusters are rescinded, making every subsequence eligible for membership to the artificial subsequence clusters. These artificial clusters are used to recreate membership levels.

A multi-stage iterative process, Phase 2 Part 2 begins only allowing Action 2 to be used on artificial clusters in which the merger size=max(MS_i), denoted MS_{max}. Just as in Phase 1, the nearest subsequences to the actionable artificial clusters are determined using Euclidean distance, and the bitsave associated with Action 2 on each cluster/subsequence pairing is computed. The greatest bitsave is then chosen, and if this bitsave is positive the associated action is taken completing the current iteration. This process continues until the

greatest bitsave is negative, or maximum number of iterations is achieved (set by user, based on time constraints).

Upon termination of the iterative process for all artificial clusters with motif size equal to MS_{max}, the iteration count is reset to 1 and Action 2 is allowed only on artificial clusters with motif size equal to MS_{max}-1. Using this new set of actionable clusters, the same iterative process is run until termination. This process is repeated, each time decreasing the motif size by 1, until all clusters have been actionable.

What does this process do? It allows for subsequence membership testing on those merged clusters to ensure that the associated motifs do occur in the time series data. In addition, for those Phase 1 clusters which have similar motifs and the same length, a cluster created in Phase 2 Part 1 composed from those two clusters, with full overlap, will be populated with the memberships of both Phase 1 clusters, leaving no eligible subsequences remaining for the Phase 1 clusters at the last stage of the phase. A minimum cluster size threshold in Phase 3 will remove low count subsequence clusters from consideration, thus replacing the two Phase 1 clusters in the above example with the Phase 2 Part 1 candidate cluster. An example of the usefulness of this multi-stage approach is given in Figure 2.4.

For the example time series shown below, Phase 1 constructed two clusters C_1 and C_2, both with similar motifs. Phase 2 Part 1 creates candidate cluster C_3 from Action 3 on C_1 and C_2. Part 2 Phase 2 first considers C_3 prior to considering C_1 or C_2, resulting in all 4 subsequences, previously members of C_1 and C_2, to be assigned to C_3. C_1 and C_2 will have size 0 at the end of Phase 2 Part 2 due to lack of eligible subsequences, and will be removed from consideration.

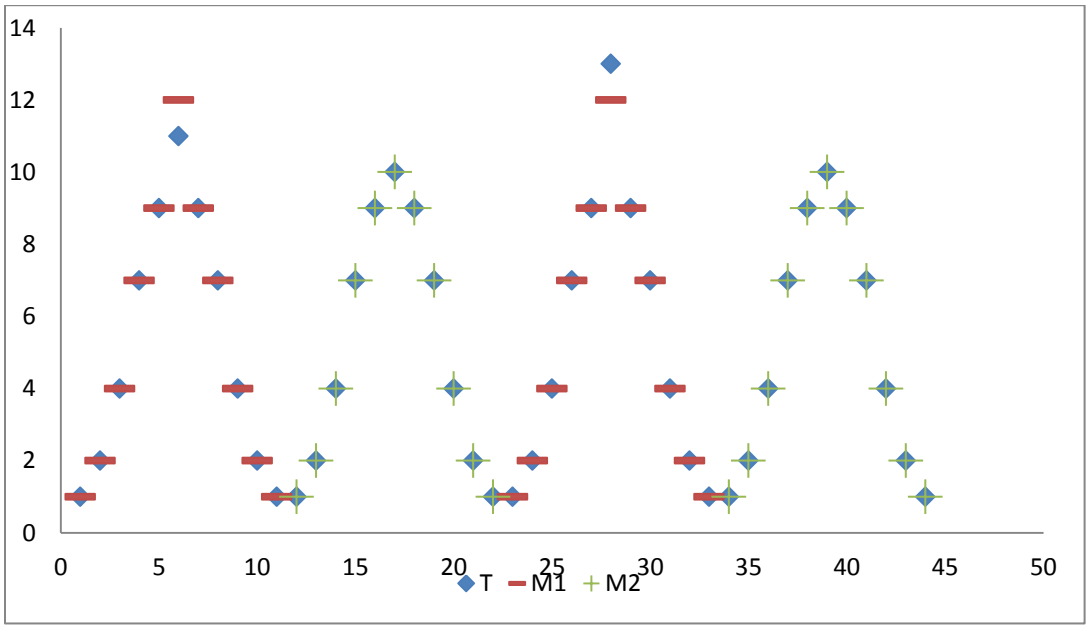


Figure 2.4 - T with associated motifs M_1 and M_2 overlaid at end of Phase 1

At the completion of Phase 2 Part 2, the members of the artificial clusters are used to recreate the subsequence clusters and associated motifs. Any clusters which have size less than a user defined threshold will be removed at the beginning of Phase 3, with the remaining representative clusters modeled by stochastic processes.

2.3.4 Phase 3-Stochastic Process Estimation

The motif discovery portion of the approach is now completed. All subsequence clusters which have size greater than the user defined threshold have associated motifs which occur frequently within the time series and are distinct. Each of these motifs can be thought of as a stochastic sub-process within the time series; each subsequence with membership a realization of that process. As a result, a hypothesized stochastic process can be determined

based upon they style of data used in the time series. The parameters of the process are then estimated using the subsequence members.

Counted data motifs may be appropriately represented by Poisson processes (Case 1). Continuous data time series may be approximated with Gaussian processes (Case 2). A third 'catch all' case, used in the presence of data not accurately represented by Gaussian or Poisson processes, is discussed briefly in this section and discussed further in Chapter 4.

Phase 3 Case 1-Counted Data

In the case that the time series variable of interest is counted data, motifs may be considered as Poisson processes, with each member of the subsequence cluster a realization of that process (Varahan 2014). If the assumptions (ii) to (iv) are satisfied or nearly satisfied (see Section 2.2.2), Poisson processes can be a useful representation of the motif. Parameter estimation of occurrence rate function $\lambda_H(t)$ is performed using spline approximations for each subsequence cluster C_H (Massey, Parker, and Whitt 1996). The function PROC TRANSREG in SAS® is used to perform this spline fitting, using cubic splines (Pedan 2014). The number of knots specified for this spline approach is given by SAS procedure guidelines (Qamar 1993). The variable of interest will be set as having a log linear response relationship with the spline function on time (Lewis, and Shelder 1976).

After creation of the occurrence rate function $\lambda(t)$ using cubic splines, an F-test is used to determine if the approximation is significantly different from the null hypothesis of a constant occurrence ($\lambda(t) = \lambda$). If the F-test evaluates the spline function (Time Nonhomogeneous Poisson process) as significant at the .1 level, the predicted values of this model will be based off this function. In the case that the cubic spline function is not

significantly different from the null model of an average occurrence rate, a time homogeneous Poisson process will be estimated.

Phase 3 Case 2-Continuous Data

Given continuous time series data, motifs could be defined by Gaussian processes. Given empirical distributions of the member values at each time epoch for a subsequence cluster, the appropriateness of using a Gaussian process can be assessed (Definition 2.14). Mirroring the approach in case 1, cubic splines are used to approximate the progression of the parameters μ and σ through the motif. F tests are used to determine if the $\mu(t)$ and $\sigma(t)$ are significantly different from the constant functions $\mu(t)= \mu$ and $\sigma(t)= \sigma$.

Phase 3 Case 3-Volatile Data

In some cases the assumptions required for Poisson or Gaussian processes are not satisfied sufficiently, and a generalized approach is used. Additionally, in cases of highly volatile motif patterns, parameter estimation via cubic spline functions or flat models is insufficient to capture the motif's pattern sufficiently. In such a case, the distribution of each time epoch of a subsequence cluster is modeled by a stochastic random variable independent of the other time epochs. This case is discussed in detail in Section 4.4.1.

Using one of the previous cases, parameter functions are estimated and a hypothesized distribution of a motif is produced. A similarity measure can now be used to determine membership of subsequences to each cluster based on these stochastic processes. Two similarity measures are chosen between, dependent on the type of stochastic process defining the motifs: chi-square like goodness-of-fit tests, and F like tests.

Similarity Measure 1: Chi-Square Like Goodness of Fit Test

The Pearson's chi-square goodness of fit test provides a statistical similarity measure for Poisson processes (Case 1). Comparing the number of occurrences at each time epoch of a subsequence to those expected from a Poisson process is used to create the associated chi-square value. Namely, for a subsequence $T_{i,k}$ and a subsequence cluster C with expected values c_j at each time epoch j, the associated chi-square test statistic is:

$$X^2 = \sum_{j=1}^{k} \frac{(t_{i+j-1} - c_j)^2}{c_j}$$

The usage of goodness of fit tests are to determine if a set of observations differ significantly from a theorized distribution with associated alpha level α. Using a larger alpha level than is normal for these tests (e.g. .3 rather than .01) only subsequences with values very near to the motifs will be defined as not significantly dissimilar from the hypothesized distribution. These subsequences will become members of that cluster.

In Cases 2 and 3, data which does not measure occurrences of an event precludes the use of such a statistical method for explicit statistical interpretation. In many applications however, variables of interest which are not explicitly counted data can still have the property of proportionality between the variance and the mean. An example of this would be sales revenue for a product where price does not vary significantly. In greater generality, if a variable of interest can be readily interpretable as a compound Poisson process, a chi-square like goodness of fit measure may be appropriate. Using the DNorm function (See Definition 2.3), data variables of this kind are transformed into integer based values. Measuring the relative magnitude of the variable of interest across a subsequence, these values are comparable to the occurrence counts used for Case 1.

While no statistical interpretation of the chi-square test statistic may be possible for this case, the pseudo p-value created does provide an accurate comparison measure for determining the relative distance of a subsequence to candidate subsequence clusters. Using a defined alpha level as a cutoff threshold, this pseudo p-value can also be used to determine membership of a subsequence to a subsequence cluster.

Similarity measure 2: F-Like Test

The chi-square goodness of fit test is a useful similarity measure when considering Poisson processes, but for many variables of interest this method is not appropriate. An F-like statistic can be created to measure the deviation of a subsequence from a stochastic process, using the associated expected variance of the process at each time epoch as a weighting factor, rather than the mean as was used in the chi-square like test statistic.

For subsequence $T_{i,k}$ and a subsequence cluster C with length k and an associated stochastic process which has expected values μ_i and variances $\hat{\sigma}^2{}_i$ at each time epoch i, the F-like statistic is:

$$F = \frac{\sum_{j=1}^{k}(t_{i+j-1} - c_j)^2}{\sum_{j=1}^{k}\hat{\sigma}^2{}_j},$$

under the null model of constant variance, with associated degrees of freedom k-1 and k-1.

As is the case for the chi-square like test, use of the associated pseudo p-value in tandem with an user defined alpha level threshold produces a low cost similarity measure which can determine the closest stochastic process defined motif to a subsequence as well as determine whether this subsequence has sufficient similarity for membership.

For the sake of convenience, references to significance of dissimilarity or hypothesis testing will assume Case 1 data, with equivalent alpha level threshold comparisons to pseudo p-values performed for data which do not use Poisson processes for motif definition.

Using the appropriate similarity measure, current members of each subsequence cluster are evaluated, removing subsequences sufficiently dissimilar from the motif. After removing these dissimilar members, eligible subsequences are evaluated against each candidate motif of the same length. If a subsequence has at least one test which fails to reject the null hypothesis, this subsequence will be considered for membership to the subsequence cluster with the least dissimilarity.

A greedy iterative approach is used to select the least dissimilar subsequence/cluster pairing, assigning membership and setting all overlapping subsequences to ineligible if at least one subsequence/cluster pairing resulted in similarity measure which failed to reject the null hypothesis. This process is repeated until there are no remaining eligible subsequences with this property.

2.3.5 Phase 4-Frequency Analysis and Similarity of Time Series

Subsequence memberships have now been assigned to each stochastic process defined subsequence cluster. For each time series, relative membership frequency to each subsequence cluster create a z-dimensional space (z=number of clusters post-Phase 3 with membership greater than some predefined threshold). A multidimensional clustering algorithm is used to group similar time series based on the feature space of motif occurrence rates. Various static data clustering approaches have been included in the code made to implement this approach.

The hard k-means clustering approach, a deterministic, Euclidean similarity measure, partition clustering approach will be used extensively in the examples given in chapters 3 and 5 (Rogers et al., 2012). Alternatively, Fuzzy C-Means, or Gath-Geva are also available for use and are preferable when there are highly different variances in the occurrence rate for the motifs (Miyamoto et al., 2008; Dumitrescu et al., 2000).

These partitions represent similar time series based on the sub-processes which act upon them. This approach provides a comprehensive subsequence-based clustering tool, providing not only the clustering for an initial set of time series, but also the motif framework for quick, low computational cost updates to these time series feature spaces. Additionally, new time series can be evaluated in short order.

2.3.6 Incremental Updates (Full and Partial)

Time series for in-process systems are not static, but require reevaluation as more data arrives. Further, a system may not be closed, with additional time series entering the set of series to be evaluated. This is exemplified in retail by new products being introduced to a store, and additional point of sales information becoming available as time progresses. In data streaming, monitor data flows in with such fluidity that storage of the entire time series is cost-prohibitive. A modified update structure using the appropriate similarity measure (see Section 2.3.4) provides a low cost subsequence cluster membership, allowing for updates to the relative motif occurrence rates as information pours in.

There is the potential of new motifs occurring as time progresses. Given a variable of interest within a business case, the number of distinct sub-processes on this time series has an

upper bound. As a result, the benefit to performing the full 5 phase approach lessens as the set of subsequence clusters encapsulates more of the sub-processes.

2.3.6.1 Partial Update

In a partial update, new membership is assessed for any new subsequences which have not been evaluated in previous steps. Using the previously estimated stochastic processes associated with motifs, the associate similarity measure tests (chi-square like or F-like) are run against the new data. A list of all eligible subsequences which had at least one test that failed to reject the null hypothesis is compiled. Using the greedy iterative algorithm, the least dissimilar subsequence will be given membership to the associated cluster, updating the list of eligible subsequences to remove those subsequences which have overlap. This process continues until the list is empty.

The feature space of relative motif occurrence is updated to reflect the new memberships and partition clustering is run to reevaluate similarity. This process requires low computational cost. If a fuzzy clustering approach is used, keeping archived snapshots of the feature space can provide insight into shifts in groups of time series, providing additional information about changes in the factors acting upon the time series.

2.3.6.2 Full update

Full process updates are run periodically to create and update stochastic processes based on new and existing subsequence clusters. Previously defined subsequence clusters are inputted into the full approach through Phase 0, with previous membership subsequences locked and removed from the input dataset to reduce computation effort (fewer iterations required in Phase 1 and Phase 2 Part 2 to achieve negative bitsave). When considering data

streams, a data stream collection process will occur periodically to retain a larger interval of data for full update purposes. Upon completion of the full update, this larger time series interval is removed from memory and the new/updated stochastic processes are used for incremental updates.

2.4. Speed and Complexity

Complexity of Actions 1-3 as well as Phase 1 are outlined in Rakthanmanon's paper on MDL-based clustering (Rakthanmanon, 2012). Let the amount of effort in evaluating the best options for Actions 1 to 3 on a specific instance be N_1, N_2, and N_3 respectively.

2.4.1 Phase 2 Part 1

Given k motif clusters from Phase 1, there will be one iteration of k(k-1)/2 evaluations of each cluster pairing. The amount of effort required for merging two subsequence clusters and evaluating the bitsave associated Action 3 is N_3. In the worst case that two clusters of minimal motif length, there are 2(MMmax-MMmin)+1 potential offsets, each of which have a cost of N_3. Therefore the first iteration of Phase 2 Part 1takes uses k(k-1)/2*(2(MMmax-MMmin)+1)*N_3 effort in the worst case.

At iteration i>1 there are k-i+1 available clusters for comparisons, and only one cluster which has not been compared to each of the previous clusters. There are k-i cluster pairings considered in this iteration, each of which have 2(MMmax-MMmin)+1 potential offsets requiring N_3 effort. The total worst case effort for iteration i>1 is:

(k-i)*(2(MMmax-MMmin)+1)*N_3.

In total there is $\left((k-1)^2 + \frac{(k-1)k}{2}\right)(2(MMmax - MMmin) + 1)N_3$ effort in the worst case of Phase 2 Part 1.

2.4.2 Phase 2 Part 2

Suppose k_2 candidate clusters result from Phase 2 Part 1. These are added to the k clusters from Phase 1, such that:

$$a_j = \#\ of\ clusters\ with\ merger\ size\ j$$

For j=1 to MS_{max}, each j-level can have up to a user defined threshold of iterations, MAXITERP2. Each iteration requires up to a_j evaluations of Action 2. Therefore the effort for Phase 2 is bounded above by:

$$\sum_{j=1}^{MSmax} a_j * N_2 * MAXITERP2$$

2.4.3 Phase 3

Given k_3 motif clusters C_I resulting from Phase 2 with sufficient size S_i, a stochastic process is fitted. The SAS package for these functions is quick, with an effort of N_4. Let N_5 be the cost of removing a sequence from eligibility in a membership. Note that $N_5<N_2$ due to N_2 including the equally effortful task of adding a sequence to membership. Membership of each subsequence belonging to a cluster is assessed using a chi-square like or F-like test. Both cases require 3n+n-1 arithmetic operations for the test statistic and a table lookup, where n is the length of the subsequence. For comparison, the Euclidean distance as a similarity measure requires 2n+n effort and is sensitive to issues such as scale.

After removal of any significantly dissimilar members from each cluster, new membership is evaluated. This process requires evaluation of each eligible subsequence against the stochastic processes associated with the subsequence clusters, requiring

approximately $\frac{3}{2}k_3N_2$ effort. Thus an upper bound on the order of effort required for Phase 3 is

$$k_3(N_4 + \frac{3}{2}N_2) + \sum_{i=1}^{k_3} S_i N_2 + lesser\ terms$$

2.4.4 Phase 4

Suppose k_4 is the number of clusters of sufficient size after Phase 3. The major computational complexity associated with Phase 4 is from the clustering algorithm. The k-means algorithm is coded using SAS® PROC FASTCLUS. The FASTCLUS procedure's exact computational complexity is not given, but may be assumed to use an algorithm such as Lloyd's Algorithm, which is of order $O(MMmax*k_4*N_T*I)$ where N_T is the number of time series being evaluated and I is the number of iterations to convergence, which is normally small. Fuzzy C-means and Gath-Geva approaches have additional complexity of calculation and Gath-Geva tends towards a higher number of iterations to convergence. Refer to Dumitrescu, Lazzerini, and Jain (2000) for further complexity discussion.

2.5 Validation Approach/Test Datasets

To demonstrate the utility of this clustering approach, test sets are created utilizing motifs occurring between segments of relative noise. The occurrence rates of motifs define 8 classes of motifs. Additional noise is overlaid on the original time series to determine the likelihood of accurate motif capturing and product grouping in response to worsening signal conditions. These test sets represent sales units data aggregated to a weekly level. This

occurrence based data variable is modeled by Poisson processes in Phase 3 (Case 1), and the chi-square test statistic is used as a similarity measure.

2.5.1 Creation of Test Time Series

Thirty five time series representing weekly sales units of products have been created. Each of these time series have a repeated yearly patterns resulting from a combination of the 5 motifs seen in the Figure 2.5 below. These patterns vary in their shape and their length. Note that some patterns such as A and B, as well as C and D are similar in shape, but the distinctness of the motifs is in their length. The breakdown of the patterns and counts of each pattern is given in Table 2.1.

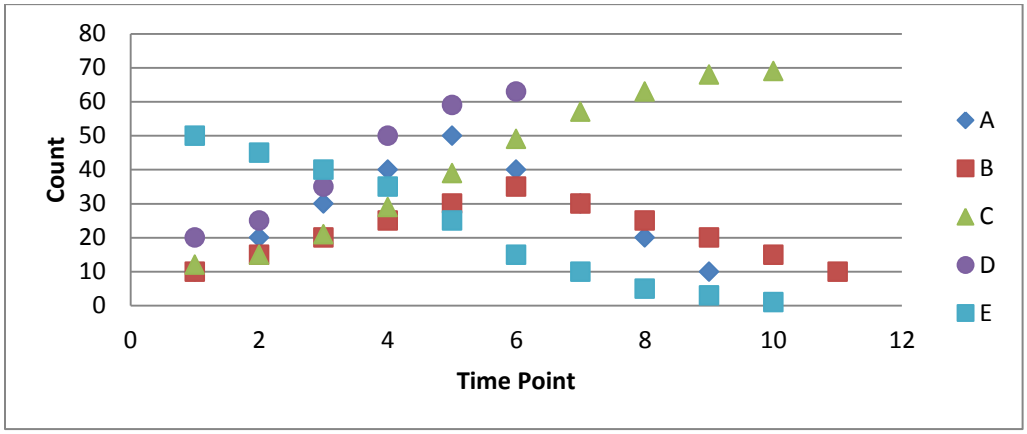


Figure 2.5 - Motif centroids for the test set

Table 2.1 - Patterns associated with time series

Pattern	Number of Series	ID Range
ACE	10	T_1-T_{10}
BDE	15	T_{21}-T_{35}
ABA	5	T_{71}-T_{75}
CDE	1	T_{76}
BCC	1	T_{77}
CEE	1	T_{78}
AAE	1	T_{79}
DEA	1	T_{80}

For the creation of these series, an interval of weeks between each motif is placed based on the Uniform Distribution on the interval [0, p] where p is 10 for all series except for BDE where it is 11. The values of the function between motifs are a rounded average of the last value of the previous motif and the first value of the future motif. The yearly pattern created from this process and the motif pattern is then copied 4 times to create a 4 year data set. To simulate differing entry/exit times as well as differing seasonality patterns for the data, truncation to the first $x_{i,1}$ weeks and the last $x_{i,2}$ weeks of time series T_i such that $x_{i,1}$, $x_{i,2}$~Uniform(0,52).

This creates a set of sales with rigid motif patterns which are identifiable. Noise is added to these time series based on the value of the realization. Therefore,

$$[T_i]_j \leftarrow [T_i]_j + f * Y_{i,j},$$

where $Y_{i,j} \sim$ Uniform(-$[T_i]_j$, $[T_i]_j$), and $f \in [0,1]$ is the fuzziness factor. Data sets with additional noise have been created for analysis with f values of 0, .05, .15, .25, and .5. These data sets are run individually through the approach with a minimum motif length of 6 and a maximum motif length of 11. The results are given in Table 2.2.

2.6 Results

2.6.1 Evaluation Criterion

The resultant outputs of the process are a set of subsequence clusters, each of which have memberships sets procured from the time series being evaluated, and the partition clustering results of those time series. The Rand index is used to quantify the level of correct classification of cluster members. The Rand index measures correct classification via pairwise comparison, with penalization for false positives and false negatives (Rand 1971).

Definition 2.14-Rand Index

Suppose a set of N elements S={$s_1,\ldots,s_N$} and two partitions of S, X={$X_{1,\ldots,}X_r$} and Y={$Y_1,..,Y_s$}. Consider each pairing of elements of S, s_i and s_j. There are 4 classifications for this pair(Rand, 1971)(Wikipedia, 2014):

1. True Positive(TP): $s_i, s_j \in X_g, s_i, s_j \in Y_u$ for some g and u
2. True Negative(TN): $s_i \in X_g, s_j \in X_h, s_i \in Y_u, s_j \in Y_v$ for $g \neq h, u \neq v$
3. False Positive(FP): $s_i \in X_g, s_j \in X_h, s_i, s_j \in Y_u$ for $g \neq h$
4. False Negative(FN): $s_i, s_j \in X_g, s_i \in Y_u, s_j \in Y_v$ for $u \neq v$

The *Rand Index* is:

$$R = \frac{TP + TN}{TP + TN + FP + FN}$$

Here X is the actual classification of the test set time series and Y is the partition clustering results of the time series using this chapter's approach. Intuitively, the Rand index is the likelihood of correct assessment of similarity between pairs of time series.

This intuitive explanation of Rand index is used to create a benchmark null model's Rand index. Given the number of classes hypothesized to occur and the number of time series present in an analysis, a strategy is produced to judge similarity/dissimilarity between pairs of time series. Given no volume information about the relative classes of time series, the assumption of equal volume between clusters is used. Using the above information a strategy is determined which creates the greatest expected Rand index. For a test set with 8 classes and 35 time series, an optimal null model Rand index of .73 is achieved. A full discussion of the creation of the optimal strategy null model is given in Appendix D. Note that the restriction to a maximum number of time series clusters, which is required for the purposes of our approach, is not a restriction in the null model. As a result, under extreme noise when subsequence pattern signal is lost, the approach outlined in this chapter could produce worse Rand index results than the optimized null model. This is demonstrated in the case of extreme levels of noise in Section 2.6.2.

2.6.2 Test Set Analysis

Five experimental data sets of time series have been created with varying levels of noise added to a signal created from differing frequencies of motifs. An example of time

series with varying increased noise is given in Figure 2.6. Each of these experimental data sets consists of 35 time series which have 8 classes. Each class has a pattern of motifs. The principal difference between time series of the same class is the differing lengths of noise between motif signals, and differing overall lengths. An example of these differences is given in Figure 2.7. These differences can cause difficulty in time series grouping using full series classification techniques.

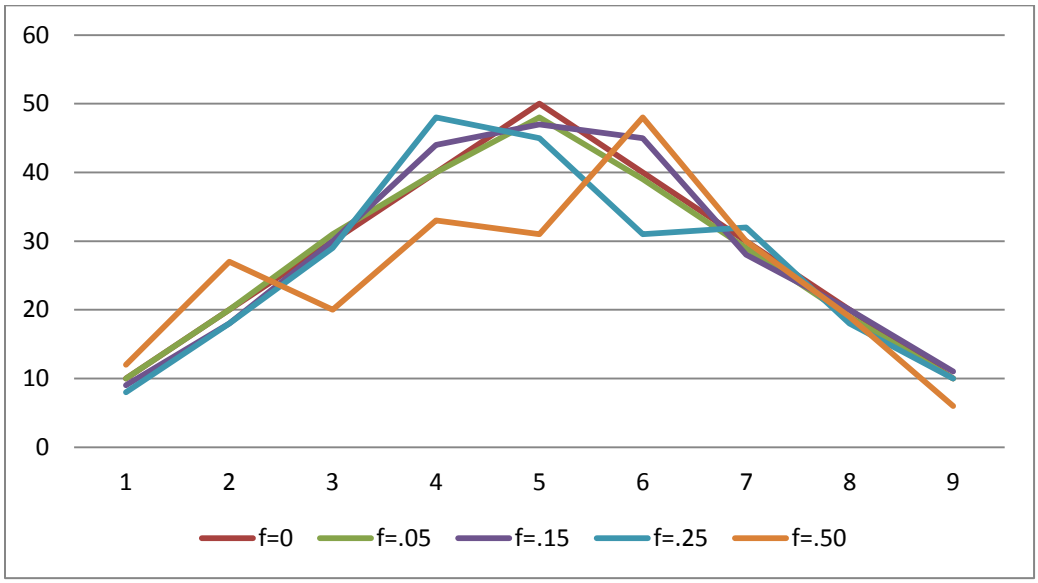


Figure 2.6 - Motif realizations under varying levels of noise

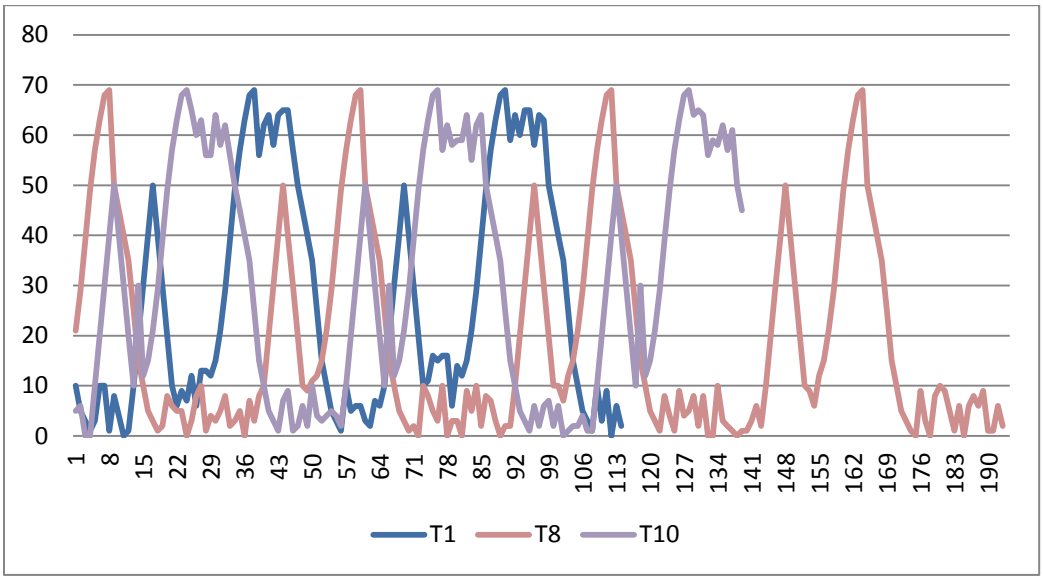


Figure 2.7 - Time series examples for same class, with f=0

Increasing the level of noise in these time series produces less clear motifs. As noise increases, the number of motifs discovered not only increases (with fewer members associated), but the quality of the motifs decreases. Figure 2.8 demonstrates this across the motifs discovered at f=0 versus f=.15.

In addition to the relative occurrence rates of motifs found in these test cases, the true occurrence rates are also run through Phase 4 as a comparison. This is the baseline test set to determine the maximal effectiveness of time series clustering using the motif occurrence feature space. The Rand Index is calculated on all 5 test sets as well as the benchmark actual occurrence rate for varying maximum numbers of classes. The results are given in Table 2.2. Hard k-means, a deterministic clustering algorithm utilizing Euclidean distance as a similarity measure, was used for time series clustering in Phase 4.

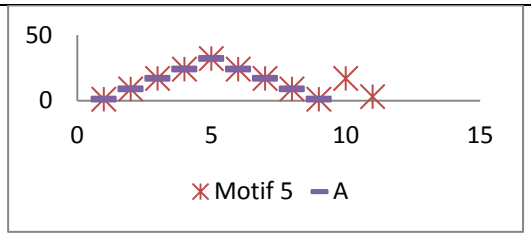
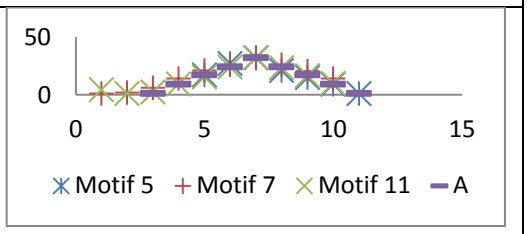
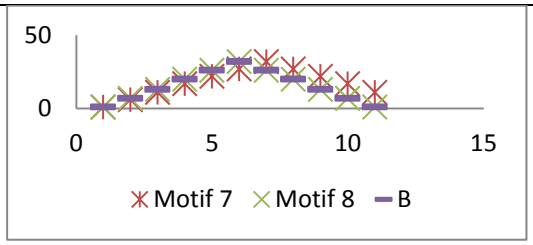
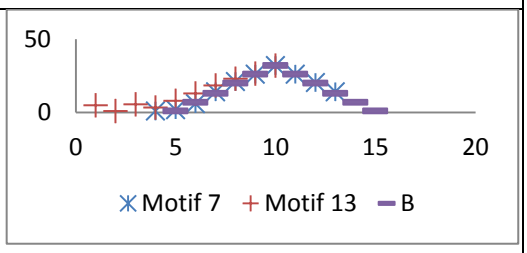
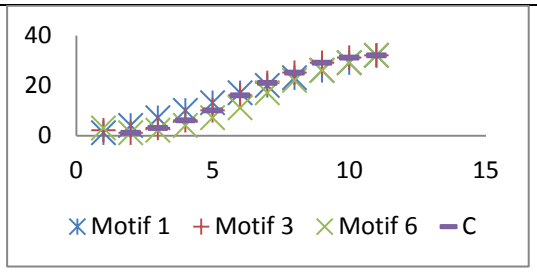
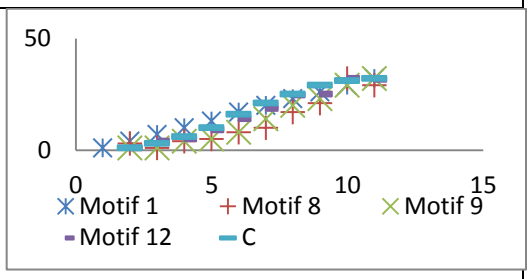
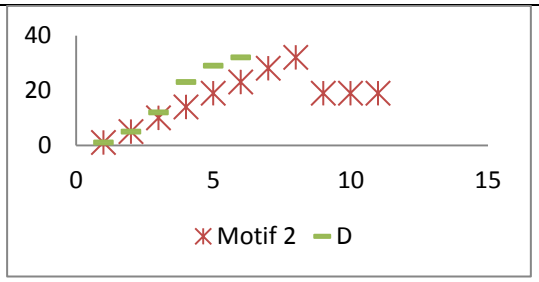
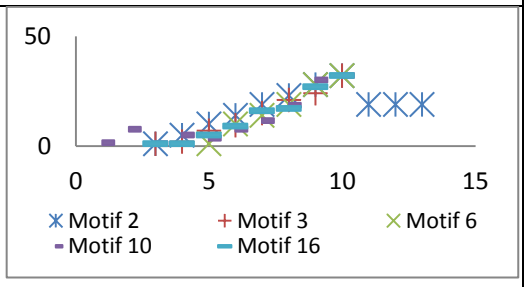
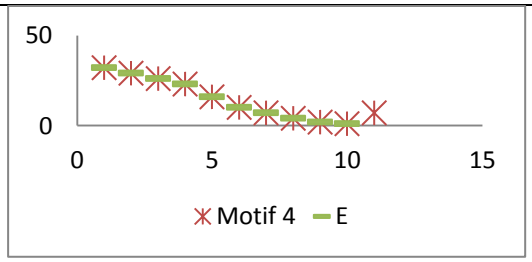
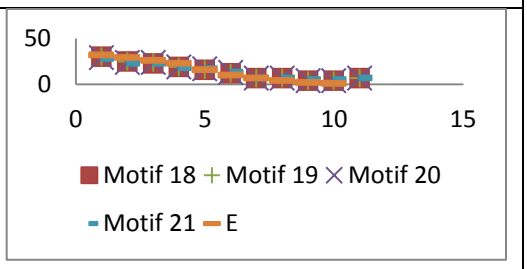
Motif	F=0	F=.15
A		
B		
C		
D		
E		

Figure 2.8 - Motif Styles by Noise Level

Table 2.2 - Rand indexes for varying noise levels and clusters.

Number of Clusters	Noise Level F					
	Actual	0	0.05	0.15	0.25	0.5
3	0.76	0.78	0.66	0.65	0.58	0.59
5	0.98	0.83	0.84	0.71	0.66	0.62
8	1.00	0.83	0.82	0.71	0.66	0.62
20	0.79	0.75	0.77	0.71	0.71	0.71

For low noise levels, the approach produced better results than the null model, with associated Rand index values of over .8 when using 5 or 8 clusters. For $F \geq .15$, the noise overlaid on the base time series created such a noisy set that motifs could not be determined in an accurate manner.

2.7 Conclusion

Time series analysis has become a topic of great interest across industries. Common subsequence patterns not only identify recurrent sub-processes acting upon the response variable of interest, but can also be used to create a feature space upon which to determine similar time series. Previous methods such as the minimum description length have provided a basis for motif discovery, but are costly and can produce spurious results as a result of greedy approaches to subsequence cluster merging. Bag-of-words, also known as bag-of-feature approaches provide quick membership evaluation, but rely on expert input for the creation of the predefined motifs. This chapter provides a methodology that creates a hybrid of these approaches for the creation of representative subsequence clusters. Stochastic

processes are defined and fitted to provide a deeper understanding of the motifs occurring in the data. These hypothesized distributions also provide a fast membership evaluation framework for incremental loads using the chi-square goodness of fit or F-like test statistic.

This approach is coded using base SAS®, and tested on artificial time series sets with varying levels of noise. Noise level growth does degrade the ability for clustering, but classification of similar time series using this chapter's approach was determined to be greater than the optimized null model in cases where the level of noise during motif portions of the time series was low relative to the intervals of no signal.

In summary, this algorithm provides a comprehensive path to classifying common time series using subsequence pattern occurrence. In the case of fractured signal due to sub-processes acting upon a data set, this algorithm has been shown to perform with great accuracy. Use of chi-square like or F-like tests during incremental loads of data provides a quick membership identification framework which can be utilized in cases of data streaming. Examples illustrating the effectiveness of this approach are discussed in Chapter 3.

CHAPTER 3: UNIVARIATE CASE STUDIES

3.1 Introduction

The MDL/Stochastic Process (MDL-SP) approach presented in Chapter 2 provides a generalized framework for analyzing univariate time series especially in the case of inconsistent signal. Sub-processes acting upon a time series, realized in the data by repeated subsequences, may not be periodic in nature. This non-periodicity of signal, mixed with moments of high noise can cause full time series clustering approaches to be ill-suited, taking in segments of noise with the same level of weight as significant subsequences. This chapter examines real-world time series data to demonstrate the usefulness of the MDL-SP approach.

Subsequence analysis is best suited in the case of repeated subsequence signals within a relatively noisy data set. This does not mean that a subsequence approach cannot be used effectively for other types of time series. Three data sets are considered in this chapter: sales quantity at a grocery store, stroke count for pool players during a billiards tournament, and power demand for an energy company. Each of these examples represents a distinct style of data for which subsequence-based time series clustering can be utilized.

The grocery study represents the ideal usage of subsequence analysis, with actors such as pricing and promotion affecting relatively stable sales patterns in potentially non-periodic occurrences. The billiards example exemplifies a data set which requires subsequence analysis, due to multiple fractured time series representing a single player with no accurate preprocessing step to reconcile missing values available. The energy study has a continuous signal data set with few to no periods of noise. This data set is apt for full time

series analysis, but subsequence analysis is shown to still be a useful tool for understanding and identifying distinct sub-processes.

3.2 Grocer Study

3.2.1 Background

Dominick's Finer Foods is a grocer based out of Chicago which partnered with Chicago Booth to conduct research on pricing and shelf-management (Chicago 2015). In an effort to provide a much needed openly-available inside look into the retail industry, store-level sales information is provided, aggregated to the weekly level with the associated pricing and promotion information. A detailed explanation and the raw data are available at http://research.chicagobooth.edu/kilts/marketing-databases/dominicks. I would like to thank and acknowledge the James M. Kilts Center, University of Chicago Booth School of Business as well as Dominick's Finer Foods for providing this openly available sales data for many categories of products across multiple stores.

Sales patterns have been a regular source of interest by companies across all industries and have resulted in extensive research (Chahrour 2010; Fok, Franses, and Paap 2007; Abdel-Khalik, and El-Sheshai 1983). Trend, seasonality, price elasticity of demand and countless other predictive components have been applied to forecast revenue dollars, profit dollars, and sales units in the retail space. Sub-processes such as pricing, promotion, and holidays act upon the rate of sales. Recurrent pricing and promotion strategies can be captured during motif discovery resulting in a feature space of sub-processes which define the lifecycle of a product.

3.2.2 Approach

Grocery data is clustered to examine the validity of two assertions.

Assertion 1: Top selling products have distinct marketing and pricing strategies from category to category.

Assertion 2: Top selling products have marketing and pricing strategies distinct from mid selling products within a category.

To test Assertion 1, 5 categories of products are chosen from the grocer: Beer, Soft Drinks, Cheeses, Cereals, and Toothpastes. In each of these categories, 3 high selling products, or SKUs (store keeping units), are chosen, and the time series of average quantity sold per store is used as the variable of interest. In order to ascertain repeated promotion patterns or heavily seasonal effects on goods, a motif window of [4, 10] has been chosen to define the subsequences for consideration. This window allows for monthly repeated single week promotions, ramp up sales, and markdown sales cycles, among others. The level of resolution, b is set to 5 (see Definition 2.3) allowing for a lossy compression to determine overall shape without oversimplification of the pattern. As this data represented occurrence of sales, Poisson processes are used to model the motifs (Phase 3, Case 1). In Section 3.2.3 it will be shown that the three beers (Miller Lite, Old Style, and Becks) were grouped within the same cluster, suggesting similarity in sales patterns distinct from other categories. To determine if this homogeneity occurs across the entire category, or just for top selling products, Assertion 2 will be examined during a secondary clustering experiment.

The examination of Assertion 2, like Assertion 1, utilizes the average quantity sold per store as the variable of interest. The scope of analysis is reduced to the single category of

beer, with a wider swath of products chosen. The 7 top selling items and 7 middle third items (mid sales items) are chosen as the time series of choice. Clustering is used to examine potential differences in composition of sales motifs based on the popularity of the product. The same motif window and level of resolution are chosen as in the previous experiment. The associated SKUs used for each experiment are given in Appendix B.

3.2.3 Results of Clustering

3.2.3.1 Average Quantity Sold for Across Categories

The creation of average quantity sold fractured the counted data of quantity into a continuous values time series. The generalized approach to Phase 3 stochastic process fitting (Case 3, see Section 2.3.4 for further detail) is used along with the additional phases in this initial experiment. No initial business user defined motifs were given prior to commencement of the approach. Ten subsequence clusters were created as a result of Phases 1 and 2 which contained sufficient size for stochastic model creation in Phase 3. The motifs associated with these subsequence clusters are given in Figure 3.1. These 10 subsequences are further stratified into patterns representative of differing parts of a product's lifecycle, given in Figures 3.2 A-C.

Phase 4 uses the final subsequence memberships to create a feature space of relative motif occurrence rates for each time series, given in Table 3.1. This feature space acts as input into a Hard-K means algorithm producing 5 distinct time series clusters, given in the 'Cluster Predicted' column of the same table. The category associated with each time series is also given for comparison to the predicted clusters.

Table 3.1 - Motif Occurrence Rate and Clustering by Product

Time Series	Motifs (Relative occurrence)										**Cluster Predicted**	**Category**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**		
Miller Lite	0	0	0	0	0	0	0	0	0	0	3	Beer
Old Style	0	0	0	0	0	0	0	4	0	0	3	Beer
Becks Regular	3	0	0	0	3	0	0	3	0	0	3	Beer
Pepsi Cola	0	3	3	0	0	0	0	0	3	35	4	Soft Drinks
Coca Cola Classic	0	0	5	0	0	0	0	0	0	25	5	Soft Drinks
Seven Up	0	0	3	0	3	0	0	0	0	30	5	Soft Drinks
Kraft Philadelphia Cream Cheese	0	0	0	0	3	0	3	0	8	13	1	Cheeses
Kraft American Singles	0	0	3	0	0	0	0	5	3	35	4	Cheeses
Dominick's Cream Cheese	0	0	0	0	0	0	0	0	0	5	2	Cheeses
Cheerios	0	0	0	3	0	0	3	0	3	20	5	Cereals
Kelloggs Corn Flakes	0	0	0	0	0	0	0	0	0	8	2	Cereals
Kelloggs Fruit Loops	0	0	0	0	0	0	0	0	0	13	2	Cereals
Crest TRT REG	0	0	0	0	0	3	0	0	5	20	1	Toothpastes
Colgate REG	0	0	0	0	0	5	0	3	3	15	1	Toothpastes
Colgate TRT Gel	0	0	3	0	0	5	0	0	5	10	1	Toothpastes

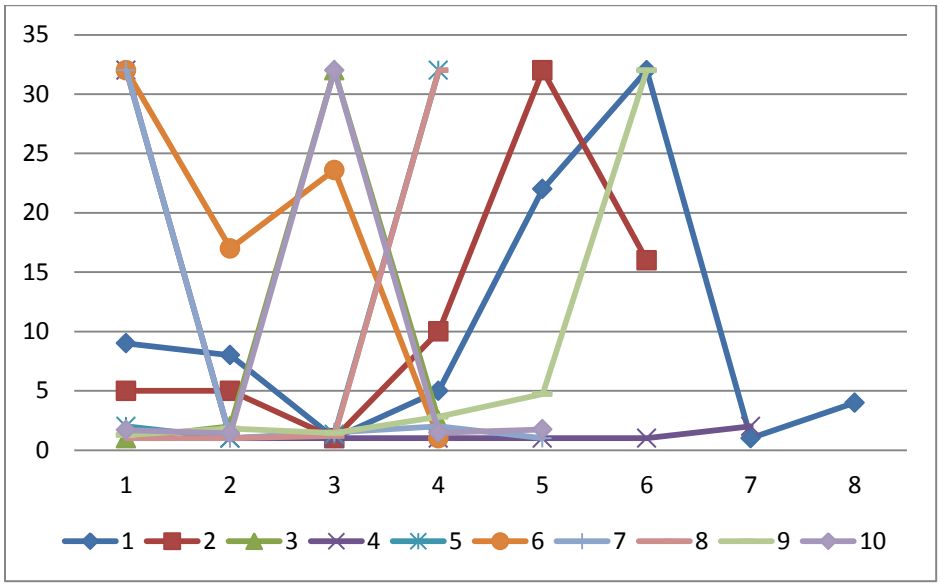


Figure 3.1 - Motifs Created from Approach

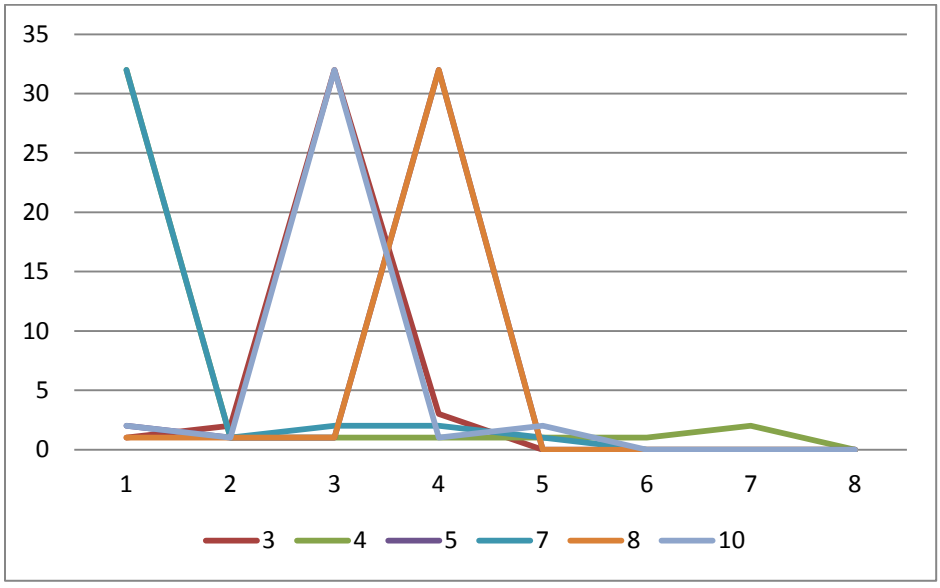


Figure 3.2A-Motifs with Single Spikes with no Ramp-up

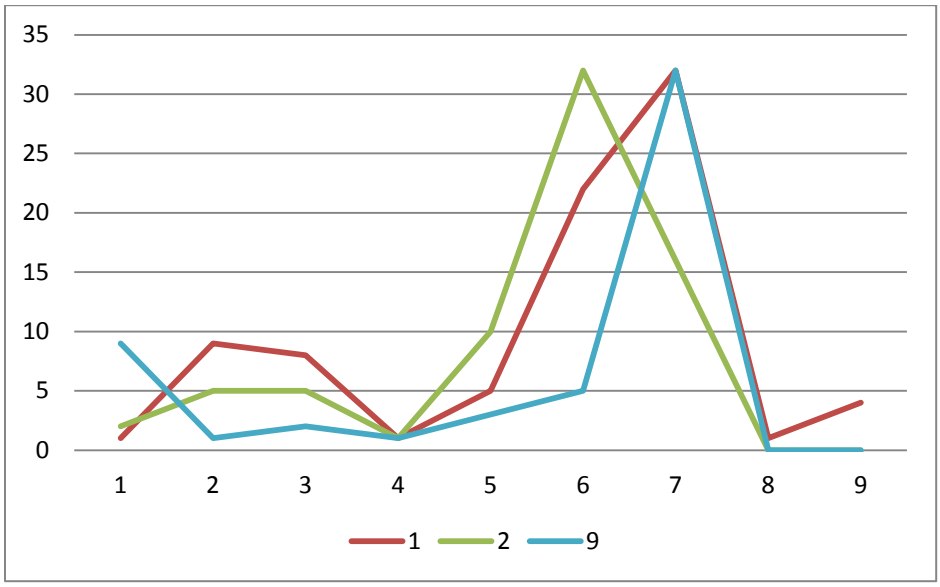


Figure 3.2B - Motifs with Ramp-up Presence

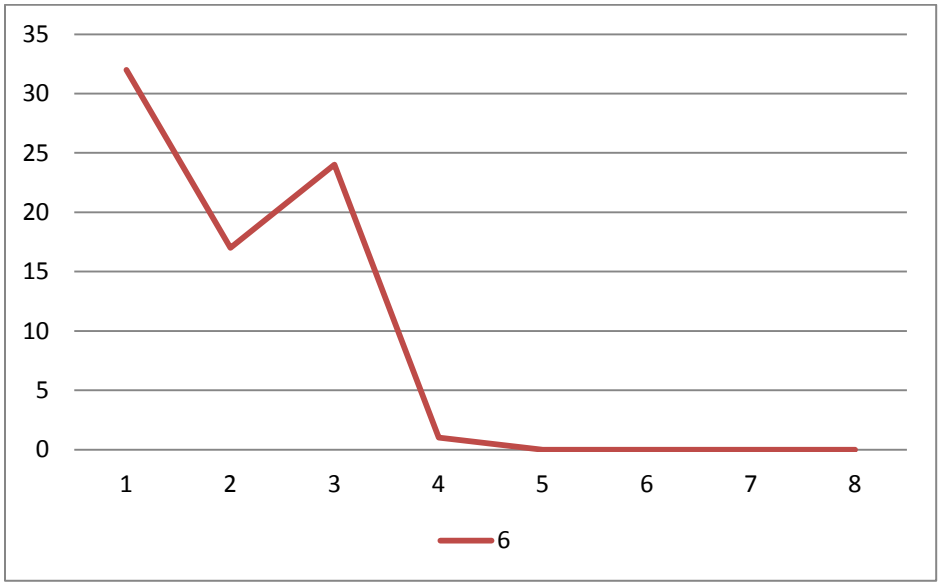


Figure 3.2C - Markdown

Cluster 3 is defined by the Beer category time series, implying time series structure for the Beer category distinct from the other examined categories. Toothpaste time series are also grouped into a single cluster, Cluster 1, but this cluster also contains Kraft's Philadelphia cream cheese. Examining the occurrence rates of the motifs for each category demonstrate distinct identifiers, bolstering the validity of Assertion 1. Motif 6 only occurs for toothpaste time series, and further occurs for all three of those time series. Beer time series do not have any occurrences of motifs 9 nor 10, which show up frequently all other categories. Soft drinks have many occurrences of motifs 3 and 10.

Clustering on these distinctive motif occurrences leads to a Rand index of .76 (using actual similarity as the category to which they belong). The null model approach results in a Rand index of .86, always choosing dissimilarity for pairwise comparison (see Appendix D for more details). This suggests that Assertion 1 does not hold across all categories. Note that this does not mean that there cannot be distinctive patterns for particular categories, as is the case with beer and to a partial extent Toothpastes (motif 6 occurrence). Further analysis of the beer category is given in Section 3.2.3.2.

3.2.3.2 Average Quantity Sold for Top versus Mid Selling Beer

Expanding on the analysis of the distinctive Beer category, an experiment is created to assess the validity of Assertion 2. Fourteen beer SKUs are chosen, divided into two classes of 7, top and medium selling products (top and mid sales). The products selected for this exercise are given in Appendix C. The same motif window and level of resolution are chosen as in Section 3.2.3.1, creating two clusters as hypothesized from the assertion. For each product, the motif occurrence rate and cluster predicted are given in Table 3.2. The

motifs found from this experiment are given in Figure 3.3, broken into groupings of similar motifs in Figures 3.4 A-D.

Table 3.2 - Motif Occurrence Rate and Clustering by Product

Time Series	Motifs (Relative occurrence)										Cluster Predicted	Selling Tier
	1	2	3	4	5	6	7	8	9	10		
MILLER LITE	0	4	0	0	0	9	0	0	9	0	2	TOP SELLER
MILLER GEN DRFT	0	0	4	4	0	0	0	0	0	0	2	TOP SELLER
OLD STYLE	4	13	4	0	0	18	0	0	0	0	1	TOP SELLER
BECK'S	3	3	0	0	0	10	0	0	0	3	2	TOP SELLER
HEINEKEN	0	0	6	0	0	0	0	0	0	0	2	TOP SELLER
SAMUEL ADAMS	3	0	0	0	0	0	0	0	0	0	2	TOP SELLER
MILLER SHARP'S	3	6	0	3	0	3	0	0	0	0	2	TOP SELLER
PILSNER URQUELL	0	6	0	0	0	3	0	3	0	0	2	MID SELLER
OREGON BREWERY	0	13	0	0	0	0	0	0	13	0	1	MID SELLER
STROHS	0	22	0	0	0	4	0	0	0	0	1	MID SELLER
BUDWEISER DRY	0	12	0	0	0	0	0	0	0	0	1	MID SELLER
MICHAEL SHEA'S IRS	5	5	0	0	0	0	0	5	5	0	2	MID SELLER
BECK'S OKTOBERFEST	0	0	5	0	5	0	5	0	0	0	2	MID SELLER
OLD STYLE ICE	0	0	13	0	0	0	0	0	0	0	2	MID SELLER

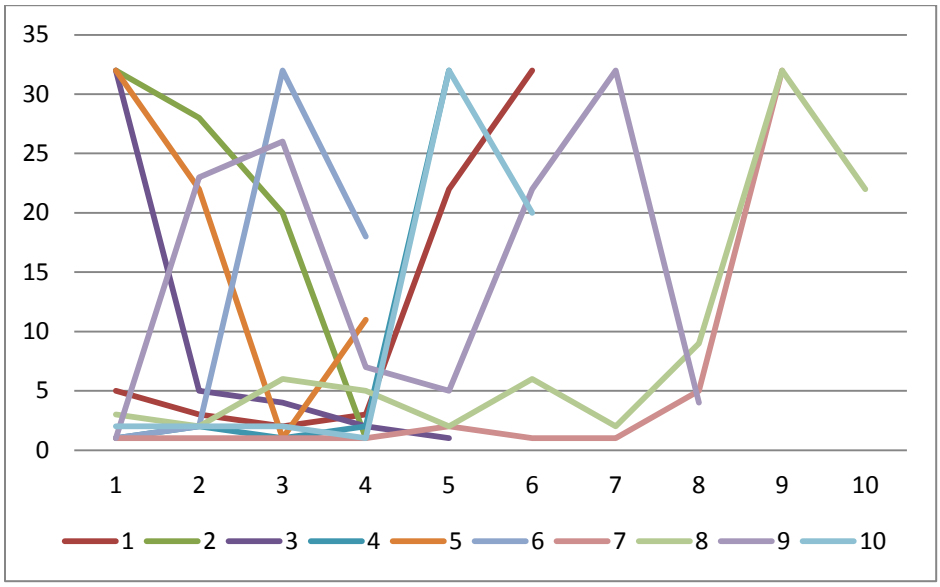


Figure 3.3 - Motifs Created from Approach

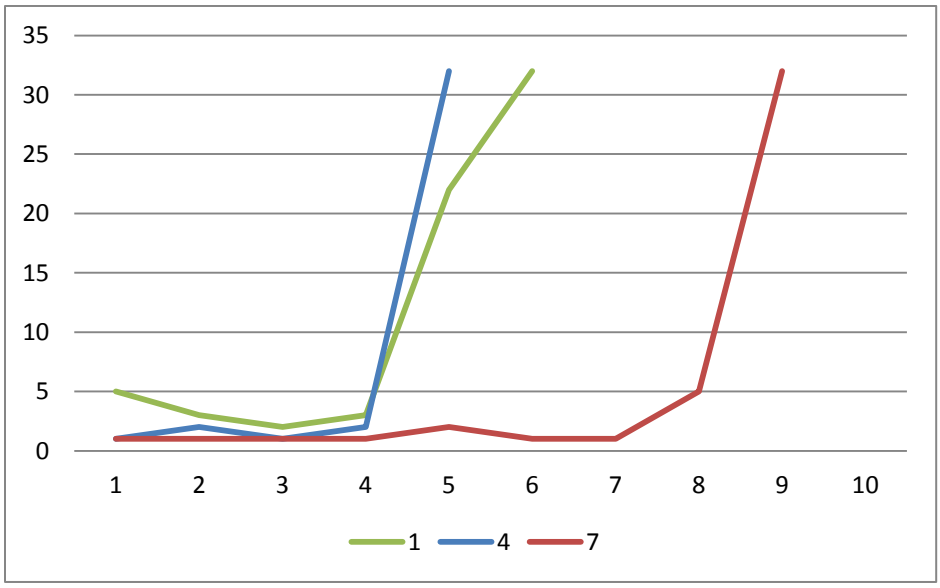


Figure 3.4A - Motifs characterized by low sales with quick peak

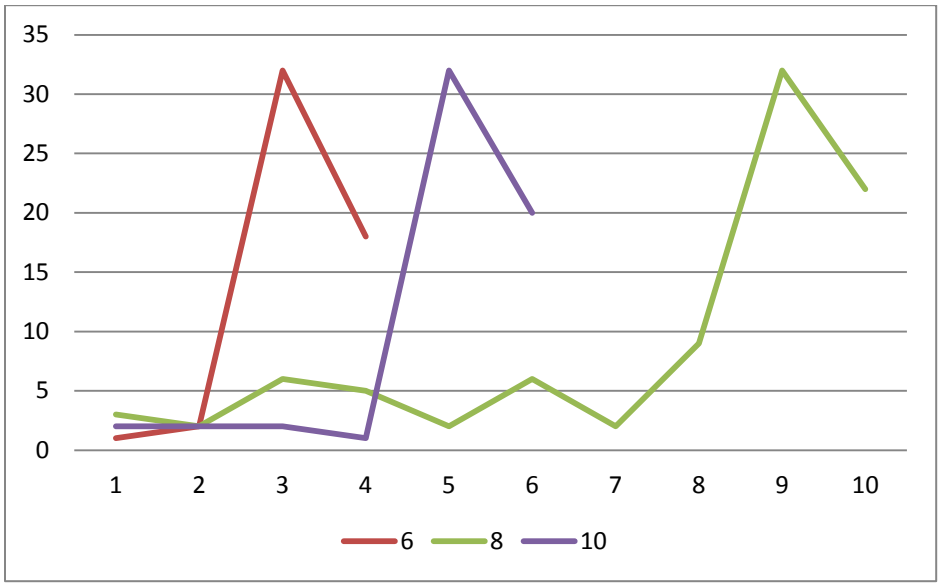


Figure 3.4B - Quick peak with a partial drop after

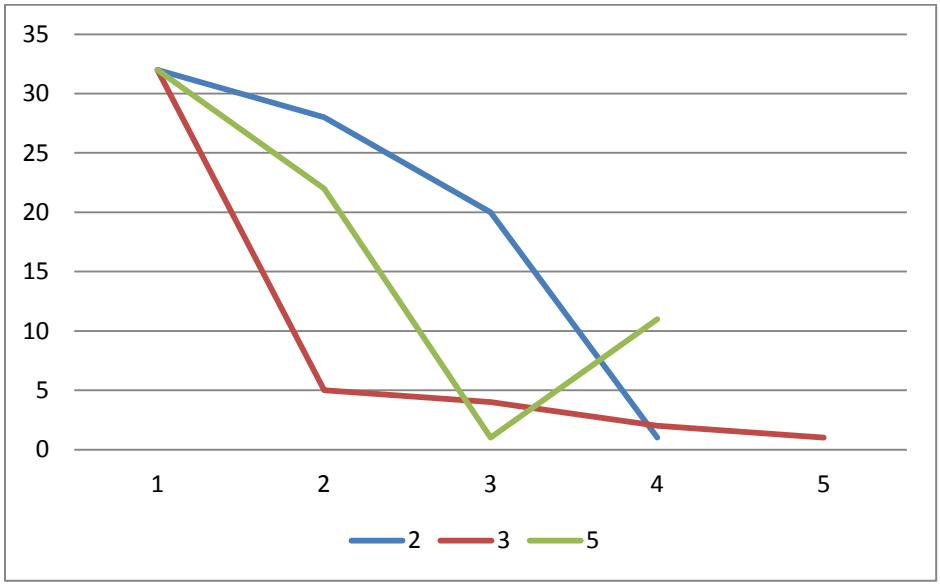


Figure 3.4C - Markdown drop

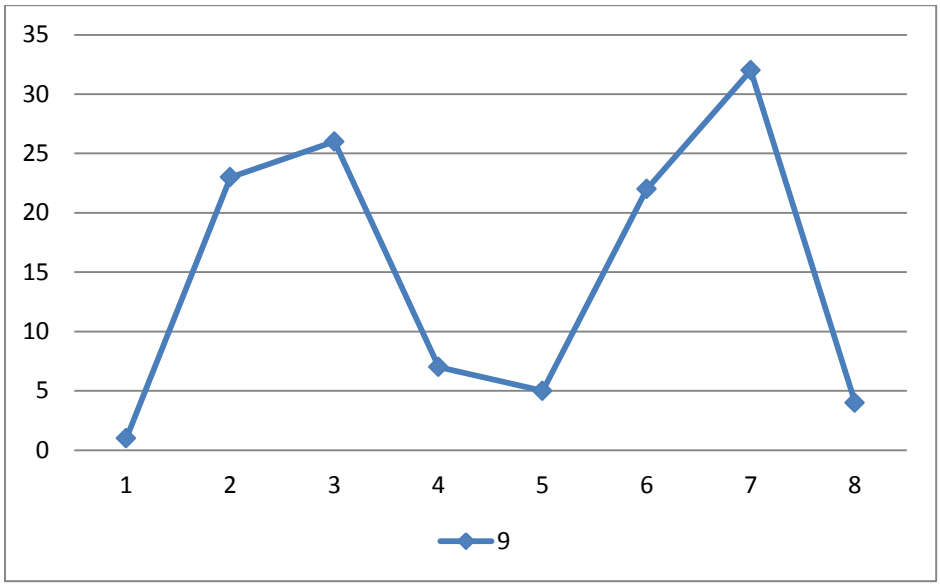


Figure 3.4D - Two-Hump Peaks

There are distinct motifs which occur in each sales tier. Motifs 4 and 10 occur only for top selling products, while motifs 5, 7, and 8 are seen only in middle sales tier products. A Rand index of .571 is resultant from the clustering, using the sales tier as the actual classification of similarity. Comparing this to the null model Rand index of .54, there is a slight increase in the predictive classification of top and mid sales tiers using subsequence clustering.

3.2.4 Concluding Remarks – Grocer Study

Two assertions were tested in the Grocer study. Theorized similarity of products by category, and market share are assessed using the clustering approach from Chapter 2. Subsequence clusters from these experiments displayed characteristics of expected sub-processes which can occur in retail products, such as promotions and markdowns.

While the Assertion 1 given in Section 3.2.2 was too strong in generality, categories such as Toothpastes and Beer displayed distinctive characteristics leading to correct pairwise comparisons with products within their categories. Assertion 2 receives some credibility in the category of beer, with a higher Rand index resulting from the clustering approach than would be expected given the null model. Given the similarity of these rand indexes, this result is weak, but distinctive low occurrence motifs were determined for each sales tier, an insight that could lead to better classification in future studies. The motifs with the greatest frequencies (2, 3, 6, and 9) have occurrences in both top and mid sales tiers, leading to the overall impression that there is a high level of homogeneity between sales patterns for beer products regardless of sales tier.

3.3 Billiards Study

3.3.1 Background

"I don't fold under pressure, great athletes perform better under pressure...so put pressure on me."

-Floyd Mayweather

In any sport, players are affected by pressure and circumstance. This is especially true when money is on the line. This pressure manifests in the game of billiards, causing inconsistent behavior, such as forgetting to chalk, and changing speed of play. This inconsistency can be time dependent, with a player forgetting to chalk more often while going on a streak of shots, to slowing play through a tournament as the number of misses accumulates. The purpose of this study is to create classes of similar players based upon response to pressure. These classes may provide strategic advantage when predicting outcomes to matches.

To create this player analysis, an 8-ball pool tournament was conducted in January 2015. Sixteen players, ranging in a variety of skill levels, played in a handicapped double elimination tournament. Each match was video recorded, with variables extracted on the presence of cue chalking, number of strokes per shot, number of shots into a player's turn, game the player was in, etc. A full list of match results and player information is given in Appendix E.

The results of this tournament provide fractured time series for analysis. The camcorders used for the data collection had short battery lives, resulting in loss of video feed during portions of the matches. This issue can occur for any monitoring system, and can hinder the use of full time series analysis. Many methods for missing value imputation have been created, but the inability to determine the elapsed time of the 'black outs' precludes such methods from use.

A benefit of the approach given in Chapters 2 and 4 is that subsequence analysis can operate on such heavily segmented sets, determining motifs across all segments. Each segment will be considered an individual time series for analysis, with summation of motif frequencies across all segments by player, occurring at the beginning of Phase 4 (see Section 2.3.5). This modification of Chapter 2's approach allows for the effective analysis of fractured time series data sets without the use of imputation to connect the segments.

3.3.2 Approach

To analyze a player's response to pressure, multiple variables were compiled during the video analysis. For the purposes of this study, the number of strokes taken by a player prior to shooting a shot will is examined. As a counted variable, Poisson processes are used

for fitting in Phase 3, and the chi-square goodness of fit test is used as the similarity measure. Additional variables of interest are considered in the multivariate billiards study in Section 5.3. The number of shots into a recorded segment is used as the time variable.

The stroke count variable has a low count rate, ranging between 1 to 31 strokes. Stroke count can be affected by factors other than pressure, such as difficulty of shot, alcohol consumption, etc., which can lead to an inherently noisy variable. As a result of these concerns, a level of resolution of b=3 is chosen for the analysis. To encapsulate stroke patterns in both short and long games, a motif window of [7, 16] is used. This additionally allows for the manifestation of lower confidence after a loss in the previous game. To prevent low volume classes, a maximum of 5 clusters is used in Phase 4.

3.3.3 Results of Clustering

Using the specification given above and in Appendix B, 15 motifs were created. These motifs are broken into similar styles, shown in Figures 3.5 A-D. Summing the occurrences of each motif by player at the end of Phase 3 results in Table 3.3. The Phase 4 classification of players is also given in Table 3.3. Using these classifications, the results of matches occurring between classes are tabulated in Table 3.4.

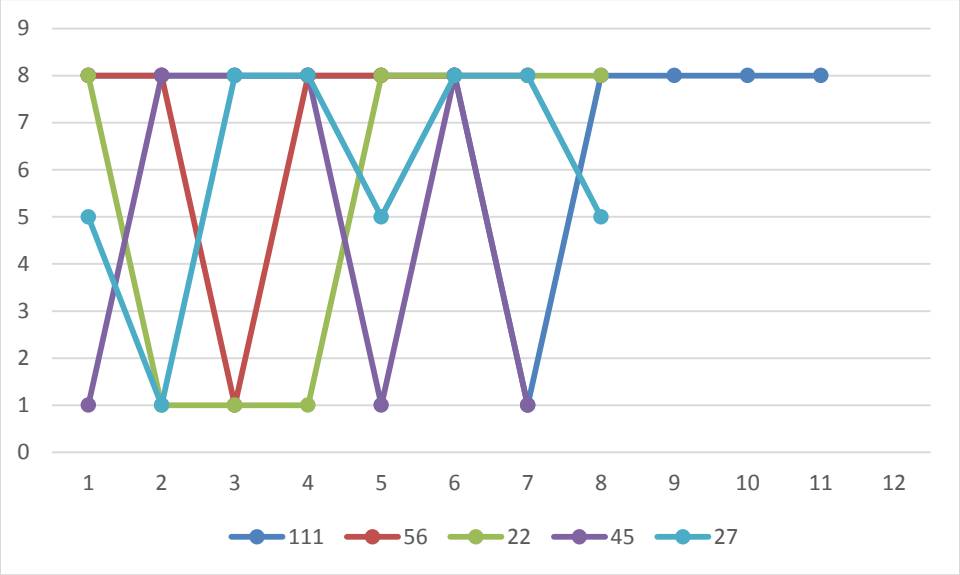


Figure 3.5A - High Stroke Standard Motifs

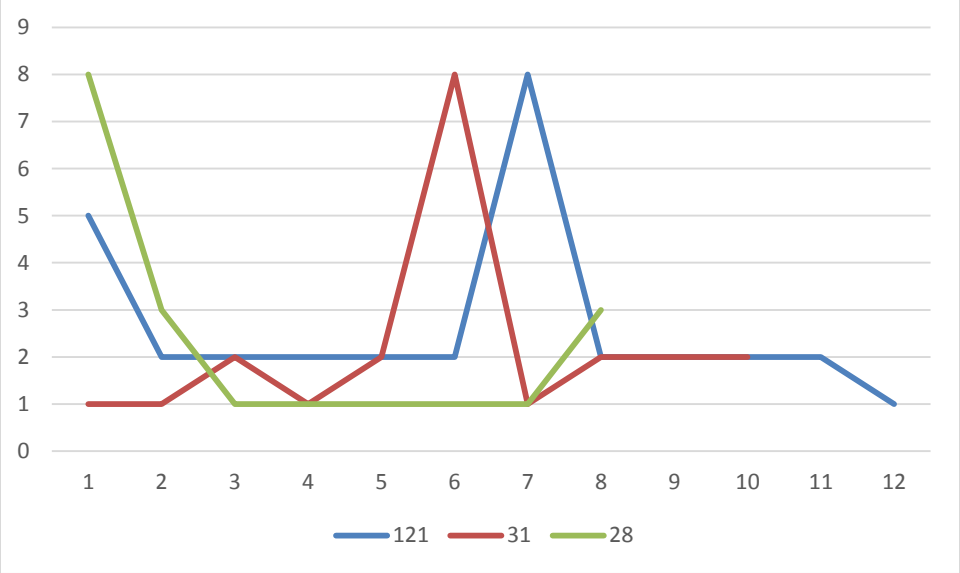


Figure 3.5B - Low Stroke Standard Motifs

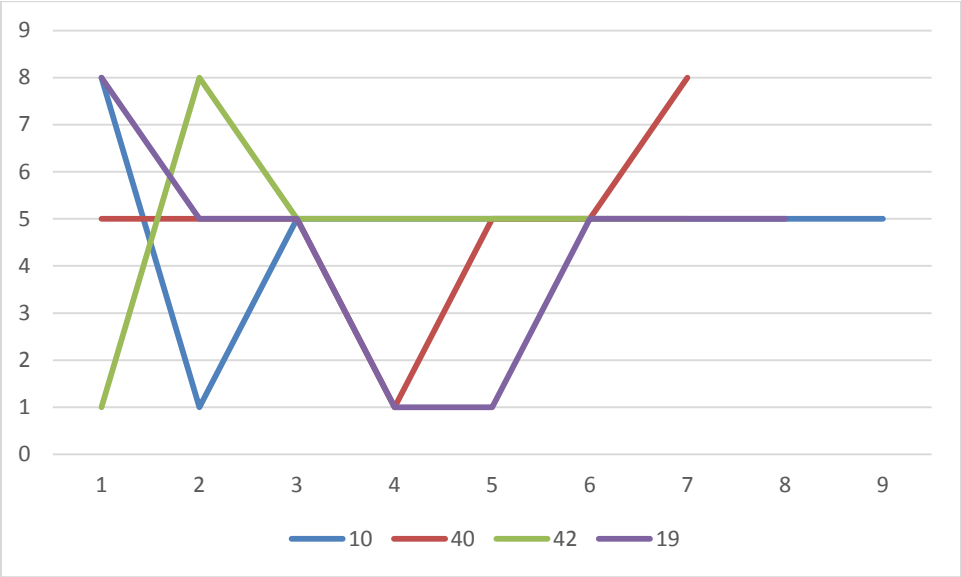


Figure 3.5C - Mid Stroke Standard Motifs

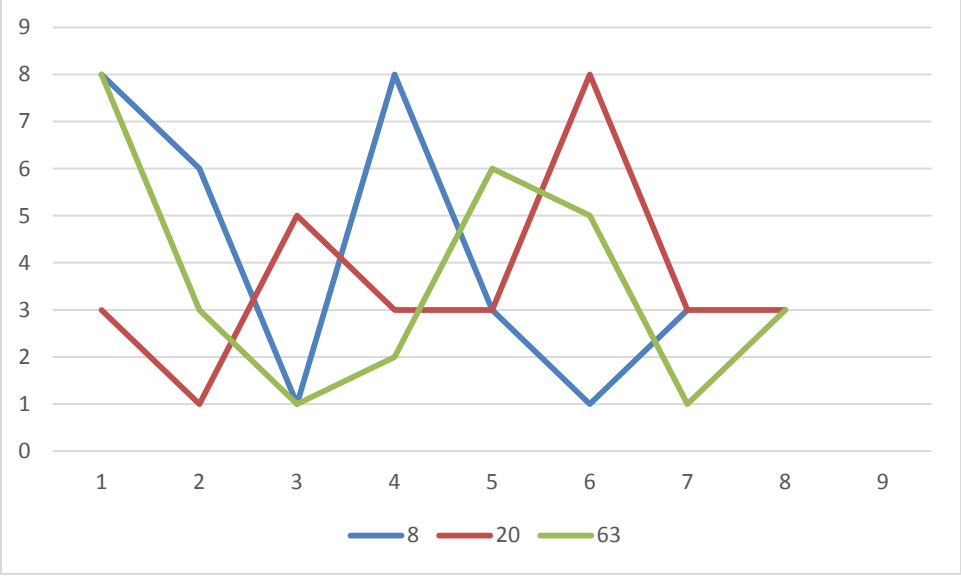


Figure 3.5D - Random Movement

Table 3.3 - Relative occurrence rates for each player

Player ID	Motifs(Relative Occurrence)															Player Class
	8	10	19	20	22	27	28	31	40	42	45	56	63	111	121	
1	2	0	0	1	1	1	0	0	0	0	0	1	0	2	0	1
2	2	0	0	0	0	0	0	0	6	2	0	0	0	0	0	2
3	1	1	0	0	1	0	0	0	1	2	1	1	0	1	0	5
4	0	2	0	0	0	0	0	0	0	2	0	2	0	1	0	3
5	0	0	0	0	0	1	0	0	1	3	0	1	1	1	0	5
6	1	2	0	1	0	1	0	0	0	0	1	0	0	1	0	3
7	0	0	0	0	1	1	1	0	1	1	0	1	0	2	0	1
8	1	0	0	0	1	1	1	0	0	1	0	1	0	2	0	1
9	1	1	0	0	1	0	1	0	2	0	1	2	1	2	0	4
10	0	0	0	0	0	0	1	0	2	2	2	1	1	0	0	5
11	1	0	1	0	1	1	1	0	1	0	1	2	0	2	0	4
12	1	0	0	0	0	0	1	1	2	1	0	2	0	0	0	4
13	1	0	0	0	0	0	0	0	1	2	2	1	0	1	0	5
14	0	0	1	0	0	0	0	0	1	1	1	1	0	1	0	5
15	1	1	0	0	1	0	0	0	1	2	2	1	1	0	0	5
16	2	1	1	0	0	1	0	0	2	2	1	0	0	1	0	5

Table 3.4 - Inter-Class Match Results

Winning Cluster	Losing Cluster	Number of Wins	Number of Matches	Percentage of Winning
1	3	3	3	100%
1	4	1	3	33%
1	5	2	3	67%
3	4	1	2	50%
3	5	2	2	100%
4	1	2	3	67%
4	2	1	1	100%
4	3	1	2	50%
4	5	4	4	100%
5	1	1	3	33%
5	2	1	1	100%

The matches in the tournament are set up using the American Poolplayers Association (APA) Equalizer® system, which compensates for unequal skill level between players by requiring additional games to be won by more capable players in order to win a match. Using their approach, the APA boasts a nearly equal likelihood of winning a match between two players regardless of skill level. (American Poolplayers Association, 2015).

Using this claim as a basis, the number of wins by a class to another class of player can be represented by a binomial distribution model with probability of success=.5. There is only .0625 likelihood of the outcome 4 out of 4 wins, which occurred between Classes 4 and 5, and only a .125 likelihood of the outcome 3 out of 3 wins, which occurred between Classes 1 and 3. Note that these calculations are based on a single pairings rather than the all interclass comparisons. Given only 6 interclass comparisons, the dominance of Class 4 players over Class 5 players could signify an inequality in performance which is not accounted for by skill level alone. Further research on this topic is necessary for this claim to be bolstered.

3.3.4 Concluding Remarks - Billiards Study

In-process systems can be imperfectly monitored. Monitoring equipment failures such as maintenance or loss of power can create fractured time series. Additionally, imputation methods to address the issue of missing values in time series may be ineffective. These resultant time series are fractured, causing full time series analysis to be inappropriate. The billiards study data exemplifies this issue. The use of Chapter 2's subsequence-based clustering techniques allows for analysis of such data.

Using the number of strokes performed prior to a shot as the variable of interest, five classes of players are discovered. Further analysis of tournament results reveal inequalities in win percentages between particular classes. Players in Class 4 beat players in Class 5 4 out of 4 times during the tournament, and Class 1 beat Class 3 all three times. These results are promising, but the low sample size of a single tournament prevents hard theories on this topic until further research is completed.

3.4 Energy Study

3.4.1 Background

"Ben Franklin may have discovered electricity- but it is the man who invented the meter who made the money."
-Earl Warren

Energy consumption, unlike a billiards tournament, is meticulously recorded. The ebbs and flows of energy demand can vary based on time of day, day of week, temperature, holidays, and countless other factors. Forecasting of electric load is a growing topic of interest, with large profit benefits to small predictive increases (Hong, 2015).

A southern Pennsylvania energy company, Duquesne light, provides service to 500,000 homes in the region (Duquesne Light, 2015). Hourly energy load data has been provided by Duquesne, available for multiple months in the fall of 2013. This data has been used in modeling approach courses in North Carolina State University's Masters of Analytics program. This study examines the validity of the following assertion:

Assertion: Weekdays are distinctly different from weekends in energy load.

The electric load data is highly cyclical, daily and yearly, especially when aggregated to the level of this dataset. This smoothness across all time suggests few to no noisy periods

between signal periods in this dataset. This conflicts directly with the key assumption of subsequence based clustering approaches that there exists insignificant noisy periods within time series. As a result, greedy subsequence based approaches to motif discovery can be susceptible to poor motif identification, selecting partial motifs which overlap, reducing the true membership count of common subsequence patterns. Modification of the original dataset using artificial fracturing provides the additional structure for subsequence-based clustering approaches to be used on such a dataset.

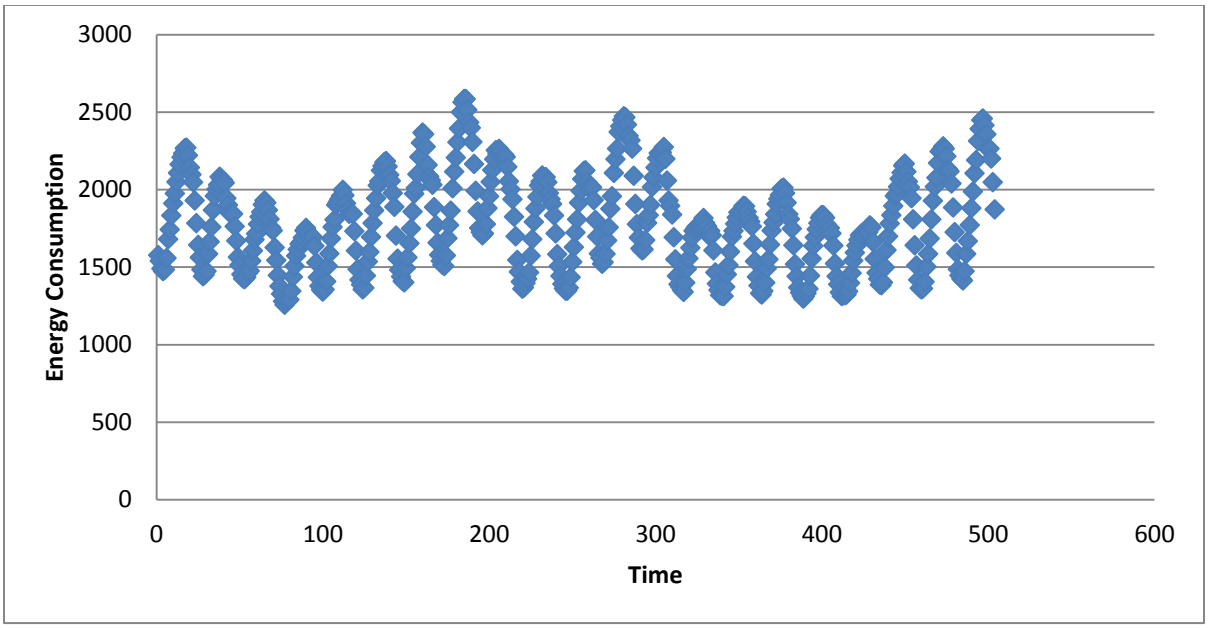


Figure 3.6 - Energy Consumption Demand

3.4.2 Approach

Approach 1: Naïve

This approach uses Chapter 2's approach to determining subsequence clusters with no modification to the original dataset prior to Phases 1-3. A motif window from 4 to 12 hours is created to allow for meaningful spikes during the day as well as more consistent periods such as night time. Case 3 of Phase 3 (see Section 2.3.4) is used, given the continuous nature of the data. The chi-square similarity measure is used. Due to the low level of noise in the data, a high level of resolution, b=6, is used.

Completion of Phases 1-3 occur on the time series. This time series is then fractured by day with the associated membership occurrences being meted out to the segments. Phase 4 is completed on these 79 segments using a maximum of two clusters. A secondary clustering schema groups motif occurrence by day of week. The relative occurrence of motifs for each day of week is then inputted into Phase 4. This approach is denoted Naïve because it is not modified to account for the continuous signal which occurs in this dataset.

Approach 2: Artificial Fracturing

The second approach segments the single time series into subseries prior to usage of Chapter 2's algorithm. Broken into individual days, each time series segment begins at 1 AM and ending at midnight. Utilizing motif lengths near the length of these segments, [20, 24], restricts each segment to at most one subsequence membership.

As is the case in Approach 1, two clustering schemas are used for Phase 4. The first uses the 79 segments separately for clustering. The second compiles motif occurrence by

day of the week prior to Phase 4. This approach artificially fractures the data, providing the structure required to effectively use a greedy MDL-based motif discovery algorithm.

3.4.3 Results

3.4.3.1 Naïve Approach- Day of Week analysis

Using the specifications given in Appendix B**,** 4 motifs are resultant from Phases 1-3, displayed in Figure 3.7**.** The subsequence clusters 78 and 88 dominated the occurrences within the original time series. Relative membership occurrence rates by date are computed, with associated time series clustering results given in Table 3.5.

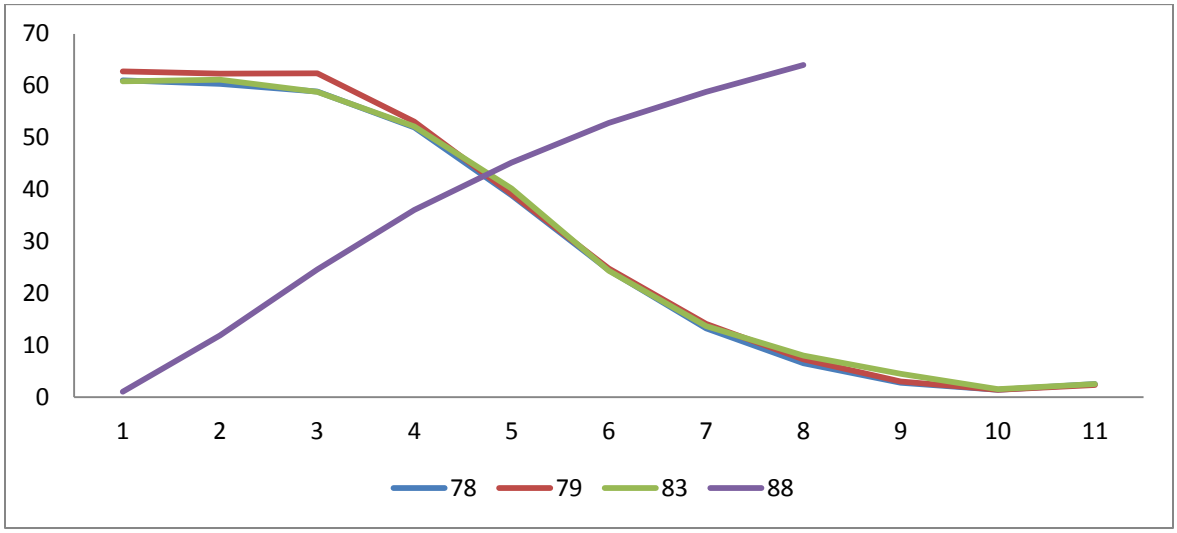


Figure 3.7 - Motifs used in Naïve approach

Table 3.5 - Relative Occurrence of Motifs by Date

Weekday	Week of Year	Relative Occurrence				Cluster Predicted
		78	79	83	88	
Sunday	**32**	6	0	0	0	1
Sunday	**33**	0	0	0	0	1
Sunday	**34**	12	0	0	0	1
Sunday	**35**	0	0	0	0	1
Sunday	**36**	5	0	0	0	1
Sunday	**37**	7	6	0	8	2
Sunday	**38**	12	0	0	0	1
Sunday	**39**	6	0	0	0	1
Sunday	**40**	0	0	6	0	1
Sunday	**41**	6	0	0	8	1
Sunday	**42**	6	0	0	8	1
Monday	**32**	12	0	0	8	1
Monday	**33**	6	0	0	8	1
Monday	**34**	12	0	0	0	1
Monday	**35**	6	0	0	0	1
Monday	**36**	13	0	0	8	1
Monday	**37**	0	12	0	8	2
Monday	**38**	6	6	0	0	1
Monday	**39**	12	0	0	0	1
Monday	**40**	0	0	12	8	2
Monday	**41**	0	0	0	0	1
Monday	**42**	12	0	0	0	1
Tuesday	**32**	6	0	6	0	1
Tuesday	**33**	12	0	0	0	1
Tuesday	**34**	12	0	0	8	1
Tuesday	**35**	6	0	0	8	1
Tuesday	**36**	6	6	0	8	2
Tuesday	**37**	0	12	0	0	2
Tuesday	**38**	0	12	0	8	2
Tuesday	**39**	6	0	6	8	1
Tuesday	**40**	6	0	6	8	1
Tuesday	**41**	6	0	0	0	1
Tuesday	**42**	12	0	0	8	1
Wednesday	**32**	0	0	6	0	1
Wednesday	**33**	12	0	0	0	1
Wednesday	**34**	6	0	6	0	1
Wednesday	**35**	0	6	0	8	2
Wednesday	**36**	0	6	0	8	2
Wednesday	**37**	0	6	0	0	2
Wednesday	**38**	6	6	0	0	1

Table 3.5 Continued

Wednesday	**39**	0	6	6	8	2
Wednesday	**40**	6	6	0	0	1
Wednesday	**41**	6	6	0	8	2
Wednesday	**42**	6	6	0	0	1
Thursday	**31**	5	0	0	0	1
Thursday	**32**	6	0	0	0	1
Thursday	**33**	6	0	6	8	1
Thursday	**34**	6	0	6	0	1
Thursday	**35**	6	6	0	0	1
Thursday	**36**	0	0	0	8	2
Thursday	**37**	6	0	0	0	1
Thursday	**38**	6	6	0	0	1
Thursday	**39**	6	6	0	0	1
Thursday	**40**	6	6	0	8	2
Thursday	**41**	6	6	0	8	2
Thursday	**42**	6	6	0	0	1
Friday	**31**	7	5	0	0	1
Friday	**32**	6	0	0	8	1
Friday	**33**	0	0	6	8	2
Friday	**34**	6	0	0	8	1
Friday	**35**	6	0	0	8	1
Friday	**36**	6	0	0	8	1
Friday	**37**	6	0	0	0	1
Friday	**38**	0	6	0	8	2
Friday	**39**	12	0	0	8	1
Friday	**40**	12	0	0	8	1
Friday	**41**	6	0	0	0	1
Friday	**42**	6	0	0	0	1
Saturday	**31**	0	7	0	8	2
Saturday	**32**	0	0	0	0	1
Saturday	**33**	6	0	0	0	1
Saturday	**34**	0	0	0	0	1
Saturday	**35**	0	0	0	0	1
Saturday	**36**	11	0	0	0	1
Saturday	**37**	6	0	0	0	1
Saturday	**38**	0	0	0	0	1
Saturday	**39**	6	0	0	8	1
Saturday	**40**	12	0	0	8	1
Saturday	**41**	0	0	0	8	2

Motif occurrence is compiled by day of week, reducing the number of time series in Phase 4 from 79 to 7. The aggregated day of week clustering results are given in Table 3.6. There were discernable difference between Wednesday/Thursday and the remainder of the week, but no demonstration of difference between weekdays versus weekends. Using weekend membership as the true classification for Rand index evaluation, there is a Rand index of value of .52 when clustering on the 79 date segments, and a Rand index of .57 when clustering by day of week segments. Comparing this to the null model Rand indexes of .41 and .48 respectively show some predictive ability in this approach.

Table 3.6 - Clustering approach Results using the Naïve approach

	Occurrence of Clustering by Date		Clustering By Day of Week
	Cluster 1	Cluster 2	
Sunday	10	1	1
Monday	9	2	1
Tuesday	8	3	1
Wednesday	6	5	2
Thursday	9	5	2
Friday	10	2	1
Saturday	9	2	1

3.4.3.2 Fractured Approach – Day of Week

Using the date as an identifier, the time series used in Section 3.4.3.1 is separated into 79 time series. A motif window of [20, 24] ensures a maximum of a single motif occurrence. These choices allow for daily motif patterns to be discovered with similar daily start and end times. Chapter 2's approach on these time series produced 3 motifs, given in Figure 3.8.

The resultant motif occurrence by date is given in Table 3.7, as well as the associated cluster predicted.

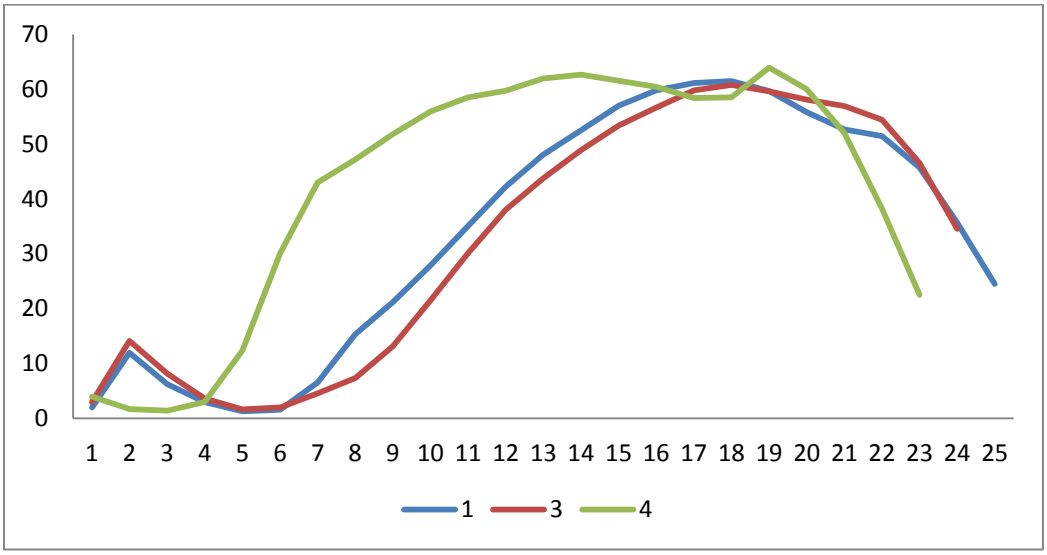


Figure 3.8 - Motif Centroid values for Fractured Approach

Table 3.7 - Occurrence rate of Motifs by Date

Day of Week	Week of Year	Motif Occurrence			Cluster Predicted
		1	3	4	
Sunday	31	0	1	0	2
Sunday	32	0	1	0	2
Sunday	33	0	1	0	2
Sunday	34	0	1	0	2
Sunday	35	0	1	0	2
Sunday	36	0	0	0	1
Sunday	37	0	1	0	2
Sunday	38	0	0	0	1
Sunday	39	0	1	0	2
Sunday	40	0	1	0	2
Sunday	41	0	1	0	2
Monday	31	1	0	0	1
Monday	32	1	0	0	1
Monday	33	1	0	0	1
Monday	34	1	0	0	1
Monday	35	0	1	0	2
Monday	36	1	0	0	1
Monday	37	0	0	0	1
Monday	38	0	0	1	1
Monday	39	0	0	1	1
Monday	40	0	0	0	1
Monday	41	0	0	1	1
Tuesday	31	1	0	0	1
Tuesday	32	0	0	0	1
Tuesday	33	1	0	0	1
Tuesday	34	1	0	0	1
Tuesday	35	0	0	0	1
Tuesday	36	0	1	0	2
Tuesday	37	0	0	0	1
Tuesday	38	0	0	1	1
Tuesday	39	0	0	0	1
Tuesday	40	0	0	1	1
Tuesday	41	0	0	1	1
Wednesday	31	1	0	0	1
Wednesday	32	0	0	0	1
Wednesday	33	1	0	0	1
Wednesday	34	0	0	0	1
Wednesday	35	1	0	0	1
Wednesday	36	1	0	0	1
Wednesday	37	0	0	0	1

Table 3.7 Continued

Wednesday	38	0	0	0	1
Wednesday	39	1	0	0	1
Wednesday	40	0	0	1	1
Wednesday	41	0	0	1	1
Thursday	30	1	0	0	1
Thursday	31	1	0	0	1
Thursday	32	0	0	0	1
Thursday	33	1	0	0	1
Thursday	34	1	0	0	1
Thursday	35	0	0	0	1
Thursday	36	0	0	0	1
Thursday	37	0	0	0	1
Thursday	38	0	0	0	1
Thursday	39	0	0	0	1
Thursday	40	0	0	1	1
Thursday	41	0	0	1	1
Friday	30	1	0	0	1
Friday	31	0	0	0	1
Friday	32	1	0	0	1
Friday	33	0	0	0	1
Friday	34	1	0	0	1
Friday	35	0	0	0	1
Friday	36	0	0	0	1
Friday	37	0	0	0	1
Friday	38	1	0	0	1
Friday	39	1	0	0	1
Friday	40	0	0	0	1
Friday	41	0	0	0	1
Saturday	30	0	1	0	2
Saturday	31	0	1	0	2
Saturday	32	0	1	0	2
Saturday	33	0	0	0	1
Saturday	34	0	1	0	2
Saturday	35	0	1	0	2
Saturday	36	0	1	0	2
Saturday	37	0	0	0	1
Saturday	38	1	0	0	1
Saturday	39	0	1	0	2
Saturday	40	1	0	0	1

Using the second clustering schema, relative motif occurrence by day of week is compiled prior to clustering. These groupings of days of week are given in Table 3.8. Using the second schema, weekdays were successfully clustered together with weekend days clustered separately. Again using the weekday/weekend classification as the true cluster membership for a Rand index evaluation, a Rand index of .82 is achieved when clustered by date, and a Rand index of 1 when motif occurrences are aggregated to the day of week. Comparing this result to the null approach with a Rand index of .59 and .52 respectively demonstrates the increase in predictive power the artificial fracturing modification can provide and validates the assertion that weekday days have different load pattern than those of weekend days.

Table 3.8 - Clustering Results for Fractured Approach

	Occurrence of Clustering by Date		Clustering By Day of Week
	Cluster 1	Cluster 2	
Sunday	2	9	1
Monday	10	1	2
Tuesday	10	1	2
Wednesday	11	0	2
Thursday	12	0	2
Friday	12	0	2
Saturday	4	7	1

3.4.4 Concluding Remarks - Energy Study

Energy load time series are continuously high in signal strength with a strong daily cyclical nature. Continuous signal time series can result in poor motif discovery when using the greedy bitsave approach outlined in Chapter 2. These poor results are due to the greedy approach assuming incorrectly that repeated subsequence patterns are broken apart by noise. Use of an artificial fracturing approach with a larger motif window creates the structure necessary to mimic noise between day segments.

A perfect classification rate when clustering by day of week, and a Rand index of .82 when clustering by date, demonstrates the advantage to using a fractured approach to the original approach (.59, and .52 respectively). Both of these approaches did beat the null model Rand indexes of .48 and .41 respectively, confirming the validity of the assertion that weekday days are distinct from weekend days. These motifs in conjunction with the goodness of fit test can be used early detection of aberration in energy load pattern throughout the day to allow for ramp up/slowdown of energy production facilities.

3.5 Overall Conclusions

This chapter presents 3 real-world case studies each representing a distinct style of time series data which can be encountered. Retail sales data in the grocer study provides a classic example of a non-fractured time series with sub-processes acting upon it at distinct times (markdowns, ramp ups, single week sales). The billiards study of stroke count provides an example of a fractured time series, in which full time series approaches to clustering are not possible, and imputation methods to create a non-fractured data set is not possible due to lack of knowledge on length of gap. Finally, the energy study examined a

continuous time series in which there was little noise across the entirety of the time series, presenting a challenge to the greedy approach of Chapter 2's algorithm. In each case, results of clustering provided insight into the time series data, and validated some of the assertions posed.

Sales quantity data from Dominick's provided a classic usage of subsequence analysis for time series clustering, in which Beer and Toothpaste products were successfully clustered together suggestive of similar sub-processes by category. For the beer category, further distinctions could not be accurately created through the clustering scheme when examining high-selling versus mid-selling beers, suggesting that sales patterns are not distinctive by popularity of item.

The stroke count data in the billiards study required subsequence-based clustering, given the heavily fractured nature of the observations. Clustering on this time series, there is a higher likelihood of Class 4 winning against Class 5 than would be expected from skill level alone. This information can be used to strategically choose players during future team tournament play.

Duquesne energy data provides a distinct difficulty to subsequence-based time series clustering methods. The continuous signal/low noise time series is apt for full series analysis, but using artificial fracturing of the time series prior to motif discover can allow subsequence-based clustering to be appropriate. This approach validated the assertion that weekdays are distinct from weekends in electricity demand daily profiles.

Chapter 2's subsequence-based time series clustering approach provides a robust, generalizable path to classifying time series data in a plethora of situations, allowing insight

into sub-processes acting upon a variable of interest, and providing actionable strategies to improve business practices.

CHAPTER 4: MULTIVARIATE SUBSEQUENCE BASED TIME SERIES CLUSTERING

4.1 Introduction

The time series clustering algorithm in Chapter 2 is shown through the applications in Chapter 3 to be a useful technique in a variety of univariate time series cases. Multivariate time series classification has an additional difficulty of determining the interplay between variables. Previous approaches to multivariate subsequence-based clustering focused on use of "stacking", in which subsequence clustering is performed on each variable of interest and then the resultant clusters are compiled (Alonso et al. 2008). While this method can provide insight into projections of multidimensional motifs onto the univariate time series, this is not a truly multidimensional analysis of subsequence patterns. Production of multidimensional subsequence patterns within time series requires an appropriate expansion of the univariate definitions, and additional approaches in usage of these motifs to classify time series.

This chapter examines complexities of a multidimensional motif definition, possible approaches, and limitations of these approaches. A hybrid-definition approach is presented to allow for a lower cost motif discovery, with more generalized subsequence classification available during incremental updates. This approach is applied to test data for validation of the concept, and additional real-world multivariate examples are given in Chapter 5.

4.2 Background and Notation

Motifs are repeated patterns in a time series. In the univariate case, motifs were defined by the contiguous subsequence pattern resultant from the average of a set of subsequences occurring in the time series data (Definition 2.7). In the multivariate case, there are multiple ways a motif may be defined. These potential definitions are based on different restrictions on the shape of a motif.

The "block" motif definition is based on contiguous subsequences. The resultant methodology can be considered a natural multivariate extension of univariate case. The "intermittent" motif definition is less restrictive, increasing the potential strength of signal for a repeated pattern, but also increasing the cost associated with subsequence comparisons in Phase 1. Sections 4.2.1 and 4.2.2 define all the required components of each approach. For these sections, assume all variables are numerical. An extension to character/nonordinal data is discussed in 4.2.3.

4.2.1 Block Motif Definitions

Definition 4.1-Multidimensional Time Series/Length

A *multidimensional time series* T is a sequence of n kx1 vectors $t_1, t_2, \ldots, t_n$ representing the realization of k variables x_m at n points in time. The *length* of the time series T is n. The notation for the value of the m^{th} variable in the i^{th} time position is $t_i[m]$. Define T[m] as the m^{th} variable time series.

Definition 4.2-Multidimensional Subsequence (Block)

A subsequence $T_{i,j} \equiv t_i, t_{i+1}, \ldots, t_{i+j-1}$ is a sequence of kx1 vectors within the ordered sequence T, of length j which is ordered as in T without exclusion.

Definition 4.3-Subsequence Cluster/Motif of a Subsequence Cluster

A *subsequence cluster* C_H is a set of block subsequences $\{SS_{H,1}, SS_{H,2}, \ldots, SS_{H,k}\}$ each of length n. A *motif H associated with subsequence cluster* C_H is the average subsequence value of the members of the cluster. The motif H is used in deviation to these specific subsequences to reduce their description length.

Using the definition of subsequence as being a contiguous sequence of j kx1 vectors, each motif will take a similar rectangular shape, creating a "block" in which each time/variable pairing (i,m) has an associated motif element defined for i= 1 to j and m=1 to k. The block structure is demonstrated in series 1 (Table 4.1) below. A length-5 repeated multidimensional pattern is marked in red. Merging subsequence $T_{5,5}$ to $T_{16,5}$ using Action 1 (described in the initial chapter) will produce a motif with positive bitsave.

Table 4.1 - Series 1

Variable	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}	t_{21}
x_1	1	4	2	6	3	5	9	5	3	3	4	8	3	5	9	5	3	2
x_2	0	2	1	6	1	1	1	1	1	5	2	0	1	1	1	1	1	6
x_3	3	4	6	2	10	9	8	7	6	2	2	6	10	9	8	7	6	5

4.2.1.1 Extension of Distance, and Entropy

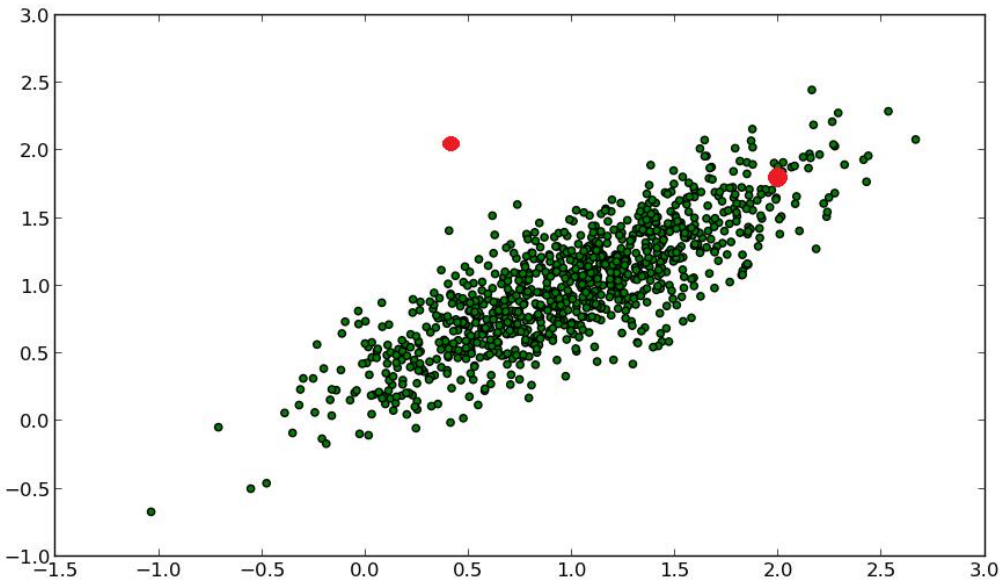


Figure 4.1 - Correlated bivariate data

Analysis of multiple variables adds the additional issue of correlation. In Figure 4.1 above, the Euclidean distances between the each point given in red, (2, 1.5) and (.5, 2), and the center (1, 1) are equal. Intuitively these points are not equivalently distanced to the center of the distribution of green points due to the correlation between variables. (.5, 2) is an outlier while (2, 1.5) is well within the possible values of this multivariate Gaussian distribution.

This example demonstrates the need to use a different similarity/distance measure in the presence of potentially correlated data. To produce a distance metric which is an extension of the Euclidean distance measure, uncorrelated variables are necessary. An

orthogonalization method on the raw data prior to the motif creation procedure produces uncorrelated factors on which a distance measure can be constructed.

Principal component analysis (PCA) is used to orthogonalize the raw data. This method, and other orthogonalization methods like it, have common use in the data mining community (Guo, Jia, and Zhang 2008; Tipping, and Bishop 2002). Use of PCA creates uncorrelated, unit variance, zero mean factors, with the added benefit that the factors are ordered descending by amount of explanatory power for the data set. To prevent sensitivity to scaling, each initial variable will be reduced to 0 mean and unit variance prior to PCA orthogonalization.

After the normalization of variables x_m, PCA transforms these variables into factors f_m for m=1, 2,…, k. The associated eigenvalue for factor f_m is denoted λ_m. The amount of variance explained in the time series by f_m is proportional to λ_m. Using these uncorrelated factors alongside the percentage total variance explained by each factor,

$$Ei = \frac{\lambda_i}{\sum_{i=1}^{k} \lambda_i},$$

it is possible to now define distance, entropy, and bitsave in a meaningful way for multidimensional time series.

Definition 4.4-Weighted Euclidean Distance

Let P(.) be the PCA transform function based on a set of time series STS. Given two equal length raw subsequences contained in STS,,r and $S_{j,r}$, such that F=P($T_{i,r}$) and G=P($S_{j,r}$), the Euclidean distance between these subsequences is defined as:

$$EDist(T_{i,k}, S_{j,k}) = \sum_{m=1}^{k} E_m \left(\sqrt{\sum_{l=1}^{r} (F_{i+l}[m] - G_{j+l}[m])^2} \right)$$

In order to approximate the entropy associated with a time series/subsequence, it is necessary to discretize the variables. Discretization will be created factor by factor on each single variable time series F[m] using the DNorm function(2.3) for associated factor y_m. Using this discretization, each factor's values in a subsequence fall within integers in the range 1 to 2^b, using the level of resolution b.

Definition 4.5-Entropy

The entropy of a multivariate time series T with associated transformed time series F is defined as:

$$H(T) = -\sum_{m=1}^{k} E_m \left(\sum_{t=1}^{2^b} P(F_m = t) log_2 P(F_m = t) \right) = -\sum_{m=1}^{k} E_m H(T[m]),$$

using the convention that if $P(F_m = t) = 0$, then $P(F_m = t) log_2 P(F_m = t) = 0$.

Description length (2.6) and bitsave (2.10) are defined in the same way as in the univariate case. Using these new definitions, Actions 1, 2, and 3, as well as Phases 1 and 2, can be defined as a straightforward extension of the univariate actions and approaches.

4.2.2 Intermittent Motif Definitions

Block subsequences are highly structured, due to the rigid rectangular definition of a multidimensional subsequence. Relaxing the definition of a subsequence slightly can allow for not all of the j x k elements to be considered. This allows for a larger set of potentially

repeated patterns to be considered. Consider series 2 below in Table 4.2. This series has a repeated pattern (outlined in red), but this pattern is not defined with the same starting point for each factor in the time series, nor the same length. This repeated pattern in the multidimensional data is also not contiguous. This motif has *holes*, defined to be elements in the subsequence which are not part of the repeated pattern. This repeated pattern is an example of an intermittent motif. In order to allow for such a pattern to be realized using Definition 4.3, it is necessary to redefine the notion of a subsequence.

Table 4.2 - Series 2

Factor	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}	t_{21}
f_1	1	4	2	6	3	5	4	5	3	3	5	4	8	3	5	6	5	3	2
f_2	0	2	1	1	1	1	1	7	1	5	4	2	1	1	1	1	2	3	6
f_3	3	4	6	2	4	9	8	7	6	2	8	2	6	4	9	8	7	6	2

Definition 4.6-Univariate Subsequence/Total Subsequence Length (Intermittent)

Let $Q=q_1,\ldots,q_j$ be a series of length j, consisting binary values _ and 1 such that $q_1=1$ and $q_j=1$. We define _ as a missing value/placeholder in the series with equivalence to 0 in math operations (i.e. a(_)=_ , a+_=a, _+_=_ for aϵ**R**). A *subsequence* $T_{i,j,Q} \equiv t_i*q_1, t_{i+1}*q_2, \ldots, t_{i+j-1}*q_j$ is a group of values within the ordered sequence of T, of *total subsequence length* j.

The requirements of $q_1=1$ and $q_j=1$ ensure that the overall length of T is retained, while including the option for exclusions within a time series (holes). Q is the *inclusion subsequence* associated with $T_{i,j,Q}$. Expanding on the univariate intermittent subsequence concept, an alternative definition of multivariate subsequences is defined.

Definition 4.7-Multivariate Subsequence/Length (Intermittent)

Let $Q=q_1,q_2,\ldots,q_j$ be a sequence of kx1 vectors consisting of binary values _ and 1 such that $q_1^T q_1 \geq 1$ and $q_j^T q_j \geq 1$ where _ is a missing value/placeholder in the series, equivalent to 0 in mathematical operations (i.e. a(_)=_ and a+_=a for aϵ**R**). A *subsequence* $T_{i,j,Q}=y_1,y_2,\ldots y_j$ is defined as a sequence of j kx1 vectors such that $y_s[m]=t_{i+s-1}[m]*q_s[m]$ and has *length* j.

Definition 4.8-Factor subsequence length

Let $T_{i,j,Q}$ be a subsequence as defined by 4.6. Factor F_m's length, L_m, on this subsequence is defined as:

$$L_m := \sum_{r=1}^{j} q_r[m].$$

If Q is given the restriction that all vectors have element values equal to 1, Definition 4.7 is equivalent to Definition 4.2. Factor subsequence length is a necessary component in the definition of this approach as Definition 4.7 does not require each factor of a subsequence to have the same length. Definition 4.8 is required when creating an appropriate definition of description length.

Definition 4.9-Description Length (Intermittent)

Let $H_m(T)$ denote the univariate entropy associated with each factor F_m of a time series T, using Definition 4.6 with excluded elements of a subsequence not used for an entropy calculation. The *description length of a time series T* is:

$$DL(T) := \sum_{m=1}^{k} E_m \, L_m H_m(T) = \sum_{m=1}^{k} E_m \, DL(T[m])$$

The description length is the weighted sum of the description lengths of each of the factor's time series. If L_m=n for all m=1 to k, then,

$$\mathrm{DL(T)} = \mathrm{L} * \sum_{m=1}^{k} E_m \, H_m(T) = n * H(T),$$

as defined by Definition 4.5. Definition 4.9 is therefore a generalization of Definition 2.6. The definitions associated with the intermittent motif produce an approach that is implementable and can provide insight into repeated pattern structure not possible when using the block structure.

Let Actions 1 and 2 be defined only between subsequences and motifs with the same inclusion subsequence Q. Action 3 (merging motifs) is defined regardless of inclusion subsequence, but with the caveat that a placeholder (excluded position) has weight 0, and a resultant inclusion matrix with value for a position equal to 1 if the overlay of the two inclusion matrices has at least a sum of one at that time epoch and factor. Using these definitions, the necessary extensions required to use Chapter 2's approach on multivariate time series is completed.

4.2.3 Character Based/Non-ordinal Data

Analysis on mixed data has been research extensively**.** Factor analysis of mixed data uses the equivalent of PCA on qualitative data and multiple correspondence analysis (MCA) for qualitative variables. Initial methods discretize continuous variables (i.e. binning) with MCA used after. Alternatively the creation of occurrence variables for each realization of a character based variable can allow for PCA to be used.

Let a variable x_i have r possible realizations, a_1 to a_r. Artificial variables $x_{i,k}$ for k=1, …, r-1 are defined as the percentage occurrence of the kth realization of x_i, i.e.

$$x_{i,k} = \begin{cases} 1 \; if \; x_i = a_k \\ 0 \; otherwise \end{cases}$$

These new variables could be alternatively defined as binary presence variables of each realization of variable x_i, but the original definition makes sense when defining a motif. Motif values for character values can now be defined by their likelihoods of occurrence for each realization. This can provide fuller insight into the nature of the motif. In the case of word recognition, this approach allows for motifs which also convey alternative words which have been grouped with a dominant word. Consider the example below of x_1 which is a character variable representing a letter in the English language, with the associated time series being a book. The motif M in Table 4.3 represents a set of two words in the same frequency of occurrence: {CAB, CAD}.

Table 4.3 - Example Motif M

Variable	$\mathbf{t_1}$	$\mathbf{t_2}$	$\mathbf{t_3}$
$X_{1,a}$	0	1	0
$X_{1,b}$	0	0	0.5
$X_{1,c}$	1	0	0
$X_{1,d}$	0	0	0.5

Prior to PCA analysis, these occurrence variables need to be normalized just as the numerical variables have been. Let the mean and variance associated with $x_{i,j}$ be $\overline{x_{\iota,J}}$ and $var_{i,j}$ respectively. Define the normalized variables to be:

$$z_{i,j} = \frac{var_{i,j}}{\sum_{b=1}^{r-1} var_{i,j}} (x_{i,j} - \overline{x_{\iota,J}})$$

The produced variables $z_{i,j}$, have 0 mean and sum variance equal to 1, if uncorrelated. This will allow for the original variable to have similar variance weighting as the numerical variables.

Computation time increases linearly with the number of factors considered. There is the potential for a large factor space in the case of non-ordinal variables with large numbers of realizations. In the case that a categorical variable does have a large set of realizations, frequency analysis can be run on this variable, with least likely realizations being binned into a single realization bucket to reduce the dimensionality with minimal loss of information. In many applications however, there are non-ordinal variables which have few realizations, such as in the case in the grocer study with promotion type as examined in Chapter 5, and creation of r-1 artificial variables is possible.

4.3 Usefulness and Feasibility of Approaches

4.3.1 Usefulness

Consider an analysis performed on series 2 if we restrict our attention to factor f_1. Series 3, given in Table 4.4 outlines the motif in red.

Table 4.4 - Series 3 - Univariate version of Series 2 for factor f1

Factor	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}	t_{21}
f_1	1	4	2	6	3	5	4	5	3	3	4	8	3	5	6	5	3	2

Using the block motif approach, the motif (3,5,5,5,3) is constructed from $t_{5,5}$ and $t_{16,5}$ using Action 1, resulting in a positive bitsave of (.39). If there are restrictions on the minimum motif length to be greater than 3 and less than 5, this motif will be chosen as the best option in the first iteration. Note that if there are no restrictions on motif length, 2 length-2 motifs, (3,5) and (5,3), will be chosen separately (bitsave=2 for each motif creation).

Now consider the intermittent motif approach. The motif (3, 5, _, 5, 3) with associated inclusion subsequence (1, 1, _,1, 1) is created using Action 1 on $t_{5,5}$ and $t_{16,5}$ with an associated bitsave of 4. This repeated pattern will be chosen and Phase 1 would terminate.

The intermittent motif approach can result in greater associated bitsave per motif.

(3, 5, _, 5, 3) has more explanatory of the repeated pattern that exists in the time series than (3, 5, 5, 5, 3). As (4.2) is a specific case of (4.6), all block motifs are considered during the intermittent motif approach.

4.3.2 Cost and Feasibility

The intermittent motif approach increases the potential for description length reduction in a time series, but also increases the cost associated with motif discovery. The chief cost increase is due to the increased number of subsequences available for comparison. In the previous example of analysis on time series 2 for factor f_1, suppose the motif window is [5, 5]. In the block approach, there are 17 length-5 subsequences, each defined by their starting position. Using the intermittent approach, for each of those 17 block subsequences, there are 8 intermittent subsequences (Q in {(1, 1, 1, 1, 1),(1,1,1,0,1), …, (1,0,0,0,1)}). Therefore there are 136 intermittent subsequences which require comparison for the creation of a subsequence cluster! For a k-dimensional time series of size n, and given a motif length L defined, there are 2^{Lk-2} intermittent series for each block series.

Action 1 for the intermittent motif approach requires the same inclusion matrix for comparison. Consequently, the lower bound of the computational effort increase for Action 1 is 2^{Lk-2}. Note this lower bound is not strict due to the intermittent approach allowing for a larger number of non-overlapping subsequences for a particular Q than would be allowed by the block approach. For example, with Q=(1, 0, 1), and Series 3, there are 16 length-3 subsequences which don't overlap $T_{1,3}$ using the block approach , and 17 length-3 subsequences which don't overlap with $T_{1,3,Q}$ using the intermittent approach, resulting in greater than a two-fold increase in the number of comparisons.

Action 2 for the intermittent motif approach does not require significantly more effort per comparison than the block approach. Using the inclusion matrix to determine feasibility of a subsequence, distance measure and entropy calculations require nearly the same effort. The only additional comparisons required are for those eligible subsequences which would have been overlapping with ineligible subsequences in the case that block motifs were used. These additional comparisons (17 vs. 16 comparisons in the example in the previous paragraph) are negligible. Similarly, Action 3 for the intermittent motif approach does not require significant additional computational effort over the block motif approach due to the inclusion matrix being defined for each subsequence cluster.

For small motif lengths and low dimensional time series, use of the intermittent motif approach can remain feasible. As the range of motif lengths and the dimensionality of the time series increases, the intermittent approach becomes prohibitively expensive due to Action 1. Using the block motif approach for full updates is cost feasible, and intermittent motifs can be created using a greedy approach for incremental updates. This approach is further explained in the methodology section.

4.4 Methodology Extensions-Full Update

In addition to the changes in definitions as defined above in the case of multidimensional analysis, addendums are required to Chapter 2's methodology to create a multivariate version of the approach. Normalization and PCA orthogonalization is required prior to beginning Phase 1. This is applied to not only the raw data but also to the business user defined motifs. Phase 3 will use Case 3 (Section 2.3.4) as described below.

4.4.1 Phase 3, Case 3-Generalized Data

Cases 1 and 2 of Phase 3, presented in 2.3.4, make specific requirements on the type of data being assessed. In Case 1 for example, Poisson processes requires counted data with the additional assumptions of Poisson processes, given in Section 2.2.2. Using the PCA orthogonalization procedure prior to Phase 1 results in uncorrelated data, but also can remove the special properties of type of data and hypothesized individual variable distributions required for previous cases. The further complication of intermittent motif discovery for usage in incremental updates creates additional difficulties in defining stochastic processes across time.

A generalized approach to Phase 3 for multivariate cases uses hypothesized distributions independently defined at each factor/time. For each factor/time in a subsequence cluster, the distribution of values from the member subsequences is used to create empirical expected values and variances.

Definition 4.10- F-Like test statistic (Block)

The *F-like test statistic* for a block subsequence G, and a dimension/length compatible subsequence cluster, with expected values $\mu_{j,m}$ and variances $\hat{\sigma}^2{}_{j,m}$ at each time epoch j for factor m is defined as:

$$\boldsymbol{F} = \frac{\sum_{m=1}^{k}\sum_{j=1}^{n}(G_j[m]-\mu_{j,m})^2}{\sum_{m=1}^{k}\sum_{j=1}^{n}\hat{\sigma}^2{}_{j,m}},$$

where k is the dimensionality and n is the length of subsequence G, and the associated degrees of freedom are n*k-1 and n*k-1. The multivariate F-like test statistic is a natural extension of the F-like test statistic presented in 2.3.4.

Beginning Phase 3, assume that there are m subsequence clusters. For each cluster C^j, $j\epsilon\{1,2,\ldots,m\}$, with members G^1 to G^r, similarity measures are assessed for each of the members. For each subsequence G^s, membership is removed temporarily from C^j creating C^{j*}. Using the F-like test and a user defined alpha level (see Section 2.3.4), the dissimilarity of subsequence G^s is assessed. If G^s is significantly dissimilar, then it is removed from the subsequence cluster C^j and the process continues with the next subsequence cluster member.

Completion of this iterative process creates refined subsequence clusters with updated motifs. F-like test statistics are created for each of the remaining eligible subsequences against these updated subsequence clusters. For each eligible subsequence, the most likely membership to a subsequence cluster is assessed. Using the alpha level cutoff, an iterative process, as used in Section 2.3.4, introduces remaining eligible subsequences with least statistically dissimilar subsequences until the alpha level is reached. Upon completion of this stage, motifs are updated based upon the subsequence cluster members. Appendix A provides pseudocode for this phase.

4.5 Methodology Extensions-Incremental Update

Time series data analysis in practice is often a continual process, updating results and analyses as more data is received (Silva et al. 2013). This continual update of time series data allows for further identification of motifs within the time series, which increases the accuracy of the time series classification. Due to time constraints for an in-process system, it can be prohibitively expensive to run the full code at each incremental update. Instead, the hypothesized distributions set at the end of Phase 3 are used as a basis for a bag of words identification approach (Baydogan et al. 2013). Intermittent subsequence clusters can be

created from the block subsequence clusters with low cost using a greedy algorithm. Usage of these intermittent motifs for incremental loads provides a sharper classification of subsequences into subsequence cluster membership.

4.5.1 Intermittent Motif Creation

At the end of the last full analysis, suppose there are r block subsequence clusters C^j with associated length L_j, and member subsequences $S_j=\{s_1, s_2,\ldots, s_{i_j}\}$. For each C^j there are $2^{L_j k-2}$ possible inclusion subsequences Q_w.

The exhaustive approach to determining the best intermittent motif, using the equivalent of Action 1 in Section 4.2, is to apply each inclusion matrix Q_w to the membership subsequences of C^j and calculate the total bitsave, $B_{j,w}$ associated with merging all i_j subsequences together simultaneously. Let Q* be the inclusion subsequence with the largest associated $B_{j,w}$ and C^j* the associated intermittent motif. C^{j*} would be the best intermittent motif for use in the incremental load process. This approach is expensive however, and a similar intermittent motif can be produced with reduced effort using a greedy algorithm.

Given the distribution of each factor/time in the block subsequence cluster from the end of Phase 3, the pairings of factor/time are ordered in decreasing order of variance. The inclusion matrix Q is initially set to all 1 (block motif inclusion), and the description length for cluster C^j (DLC_1) is assessed, using Definition 4.9 applied to Definition 2.9. Let Q_z be the inclusion matrix used for description length computation in the zth iteration. At each iteration z, the factor/time pairing with the largest variance, which is included in Q_{z-1}, is excluded to create Q_z. Using this updated inclusion matrix, the description length of the intermittently defined subsequence cluster is assessed (DLC_z). If DLC_{z-1}-DLC_z>0, there is

positive associated bitsave with the new action of removing a factor time pairing from the inclusion matrix for subsequence cluster C^j. This iterative procedure terminates when DLC_{z-1}-DLC_z<0 or Q has minimal inclusion to maintain the length of the subsequence cluster.

As each factor has unit variance due to PCA, the greedy approach removes the most volatile factor/time pairings to create a subsequence cluster which has lower subsequence member variation. At each iteration, the entropy for the cluster will be most reduced given any single exclusion. Given the reduction in the summation of the factor subsequence lengths due to the exclusion, the description length may not be reduced, causing termination of the iterative procedure.

The computational advantage to the greedy method is clear, with at most L_jk-2 bitsave computations needed, compared to the $2^{L_j k-2}$ bitsave computations required in the exhaustive approach. As a result, the greedy method to intermittent motif creation is used for incremental loads.

4.5.2 Incremental Load Update

Consider the case of s k-dimensional time series, on which the clustering approach is applied using motif window [L_{min}, L_{max}]. Single time point additions create at most s(L_{max} - L_{min} +1) new block subsequences. For each subsequence of length L, evaluation of an F-like test statistic is created for each dimension/length compatible intermittent subsequence cluster C using the definition below.

Definition 4.10- F-like test statistic (Intermittent)

The *F-like test statistic* for an subsequence G, and a dimension/length compatible intermittent subsequence cluster C, with expected values $\mu_{j,m}$ and variances $\hat{\sigma}^2{}_{j,m}$ at each time epoch j for factor m is defined as:

$$\boldsymbol{F} = \frac{\sum_{\boldsymbol{m=1}}^{\boldsymbol{k}} \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} q_j[m](G_j[m]-\mu_{j,m})^{\boldsymbol{2}}}{\sum_{\boldsymbol{m=1}}^{\boldsymbol{k}} \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} q_j[m]\hat{\sigma}^2{}_{j,m}},$$

using the convention the convention that if $q_j[m]$=_ then $q_j[m](G_j[m] - \mu_{j,m})^{\boldsymbol{2}}$_ and $q_j[m]\hat{\sigma}^2{}_{j,m} =$ _, and as a result are not considered in the summation.

There are $\sum_{i=1}^{L_j} q_i{}^T q_i$-1 and $\sum_{i=1}^{L_j} q_i{}^T q_i$-1 degrees of freedom associated with the above test statistic. Using the F-like test for evaluation of eligible subsequences, membership is added iteratively using the same procedure as the end of Phase 3, Section 4.4.

Upon completion of the new subsequence membership addition stage, the relative frequency of motif occurrence is updated for each time series and Phase 4 is rerun, using the most recent Phase 4 centroids as the initialization for the Phase 4 clustering approach.

4.6 Validation Approach/Test Datasets

To demonstrate the efficacy of this multivariate clustering approach, a set of 3-dimensional time series have been created. These time series are broken into 4 distinct classes based upon their prevalence of motifs. Additional real-world data examples demonstrating the use of this algorithm are given in Chapter 5.

4.6.1 Creation of Datasets

32 time series were created for the purpose of this demonstration. These 32 time series T_1 to T_{32} are broken into 4 classes/groups of similar motif occurrence rates:

$G_1=\{T_1, T_2,\ldots,T_8\}$, $G_2=\{T_9, T_{10}, \ldots, T_{16}\}$, $G_3=\{T_{17}, T_2,\ldots,T_{24}\}$, and $G_4=\{T_{25}, T_{10}, \ldots, T_{32}\}$. There are 6 3-dimensional motifs which are used for the composition of these test time series, displayed in Figures 4.2 A-F. The motifs associated with each of the classes are given in Table 4.5.

Table 4.5 - Motif Patterns by Class

Class	Pattern of Motifs
G_1	A, B, A
G_2	C, D
G_3	B, C, E
G_4	F, D, A

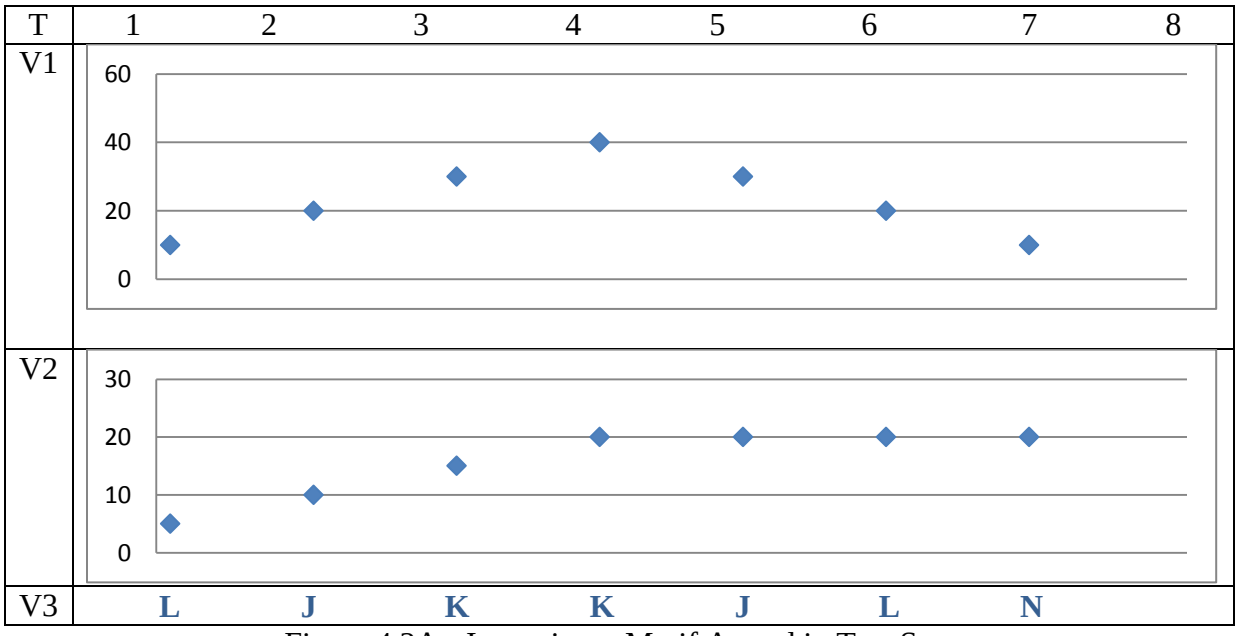


Figure 4.2A - Intermittent Motif A used in Test Sets

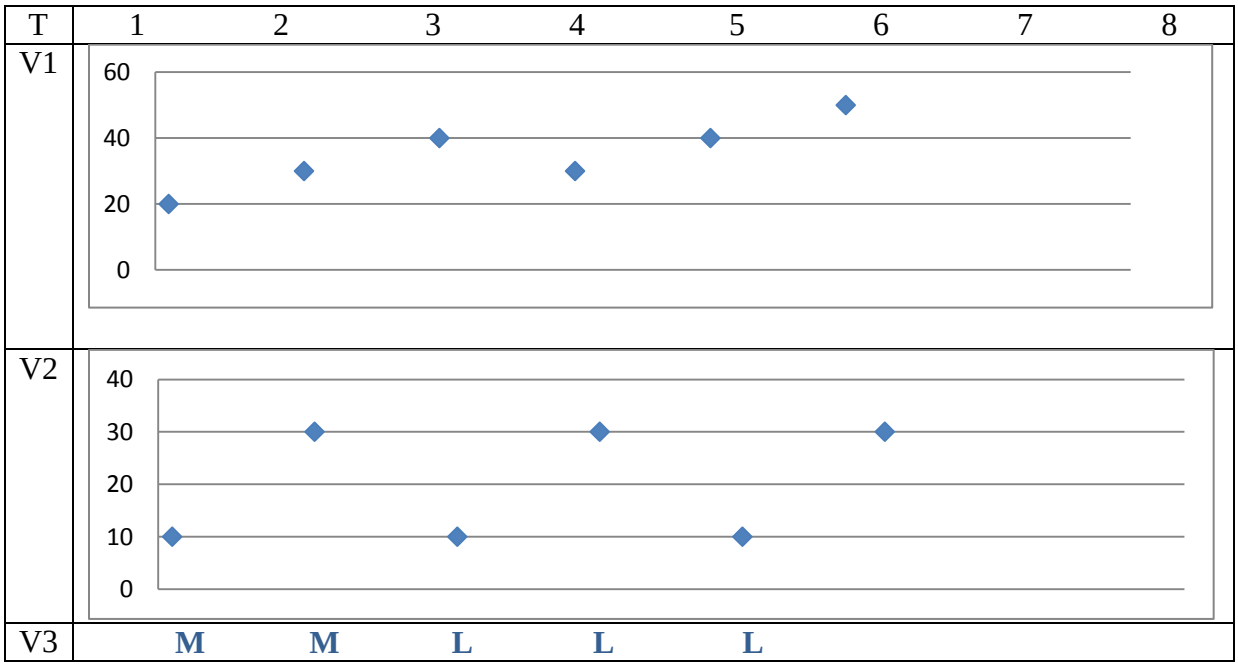


Figure 4.2B - Intermittent Motif B used in Test Sets

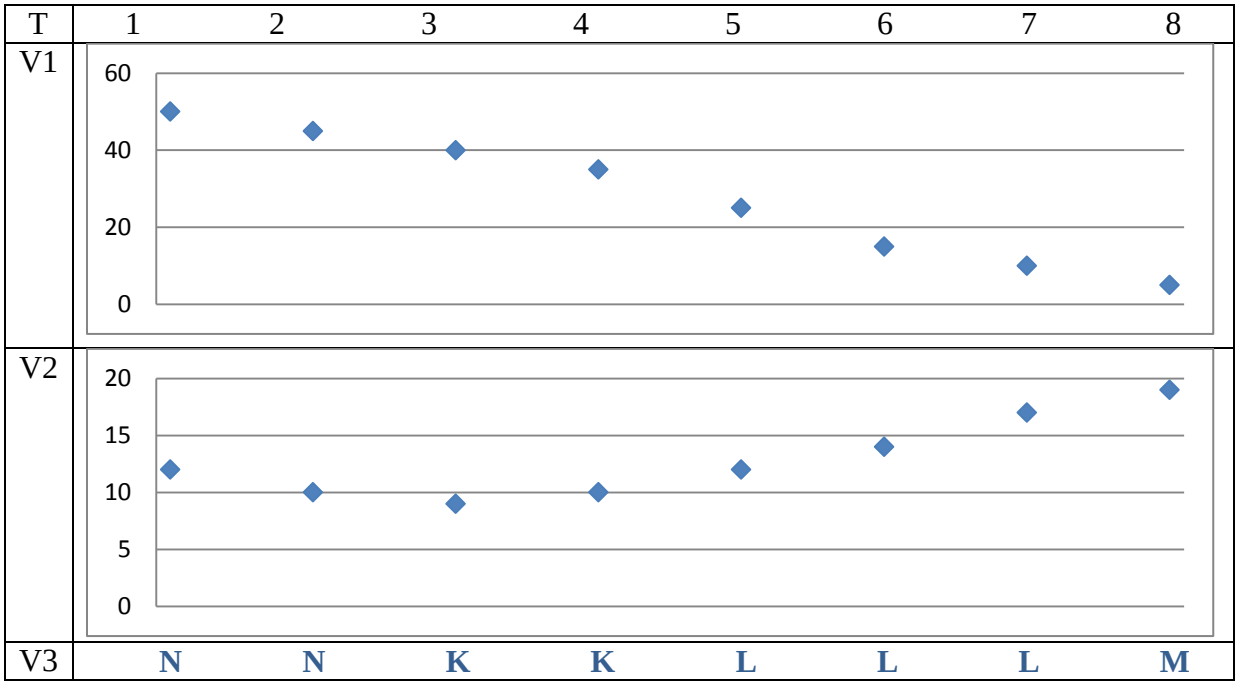


Figure 4.2C - Block Motif C used in Test Sets

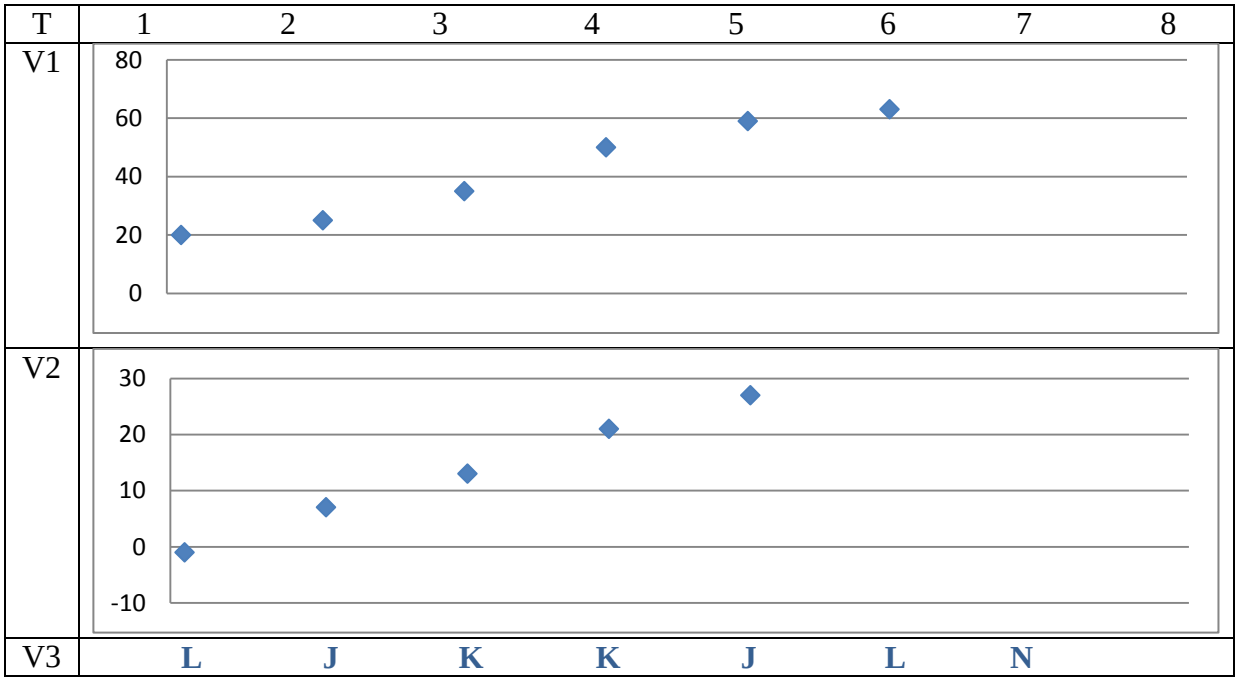


Figure 4.2D - Intermittent Motif D used in Test Sets

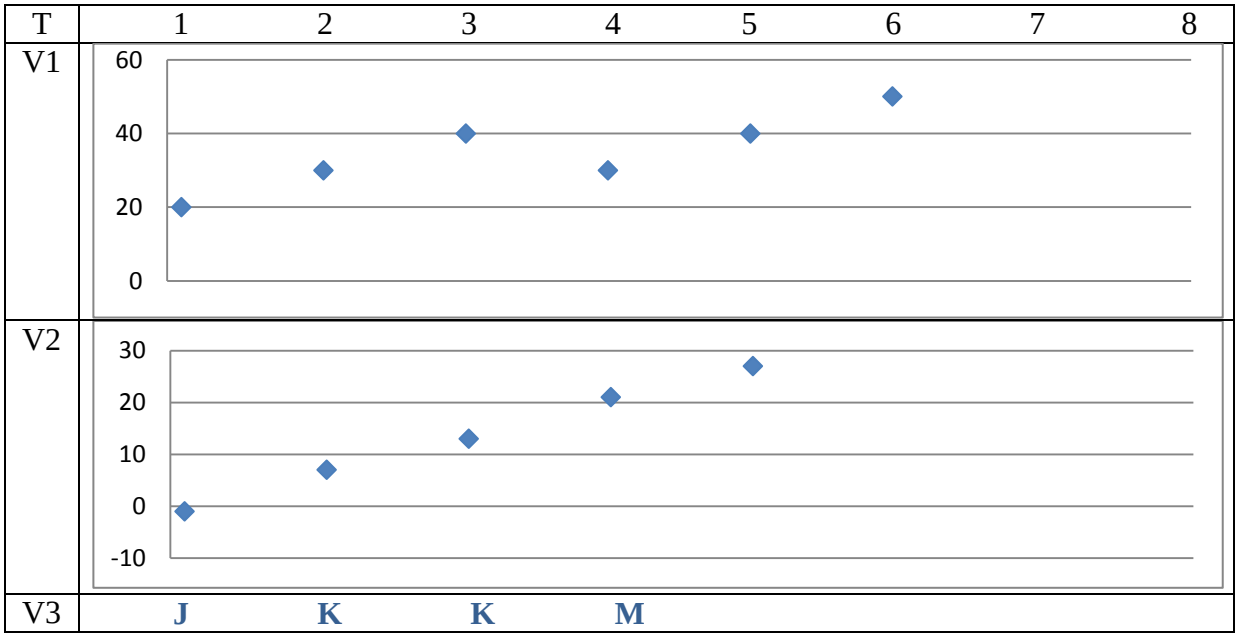


Figure 4.2E - Intermittent Motif E used in Test Sets

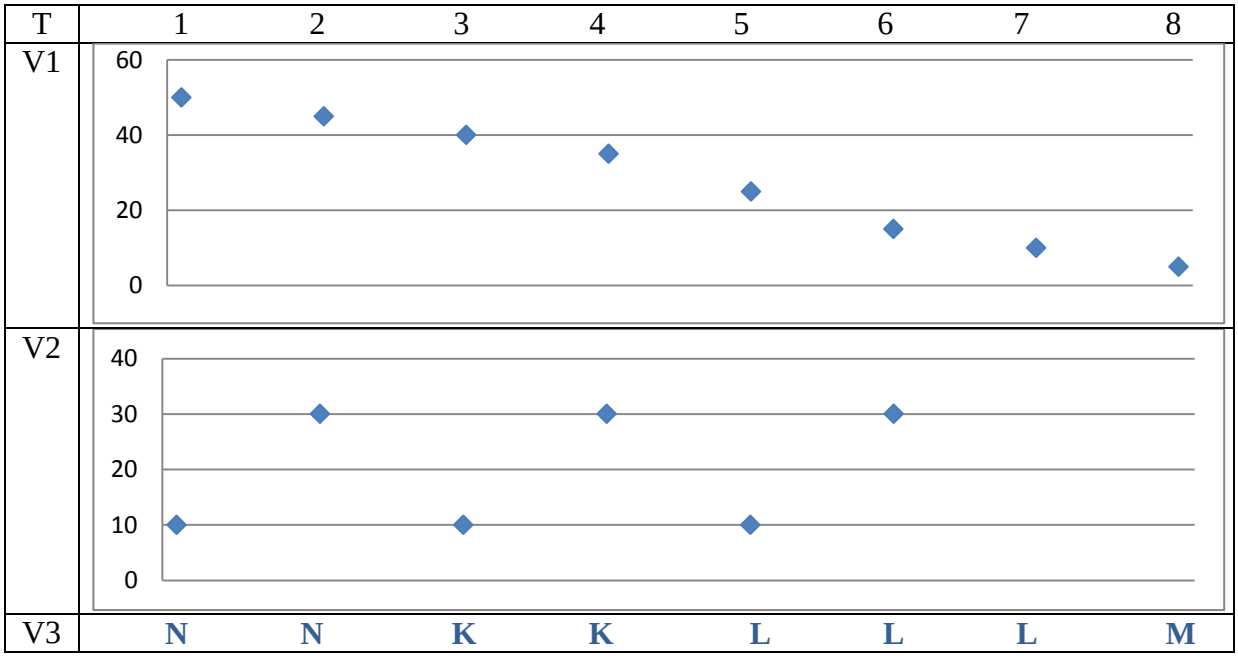


Figure 4.2F - Intermittent Motif F used in Test Sets

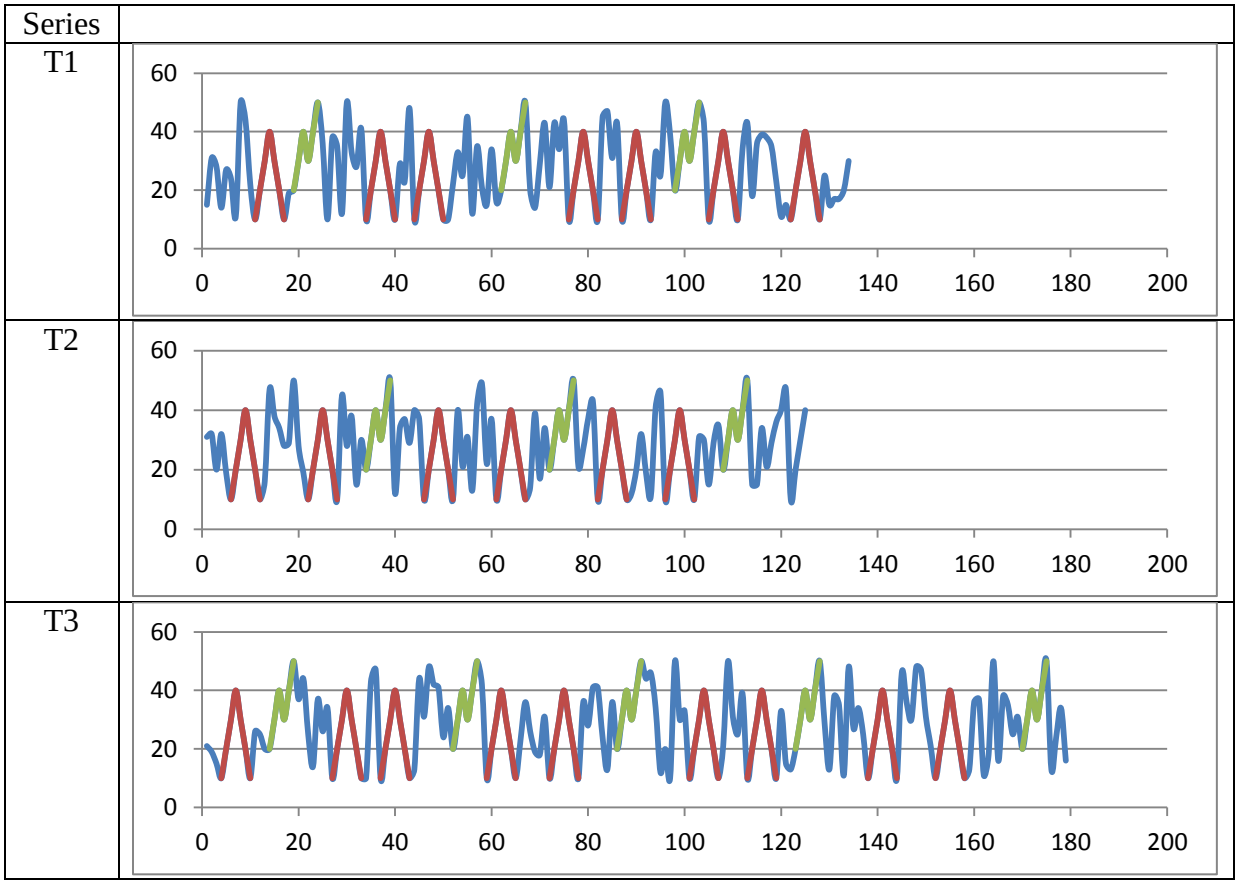


Figure 4.3A - Test set examples of C_1 for Variable 1. Motif A is in red, Motif B in green.

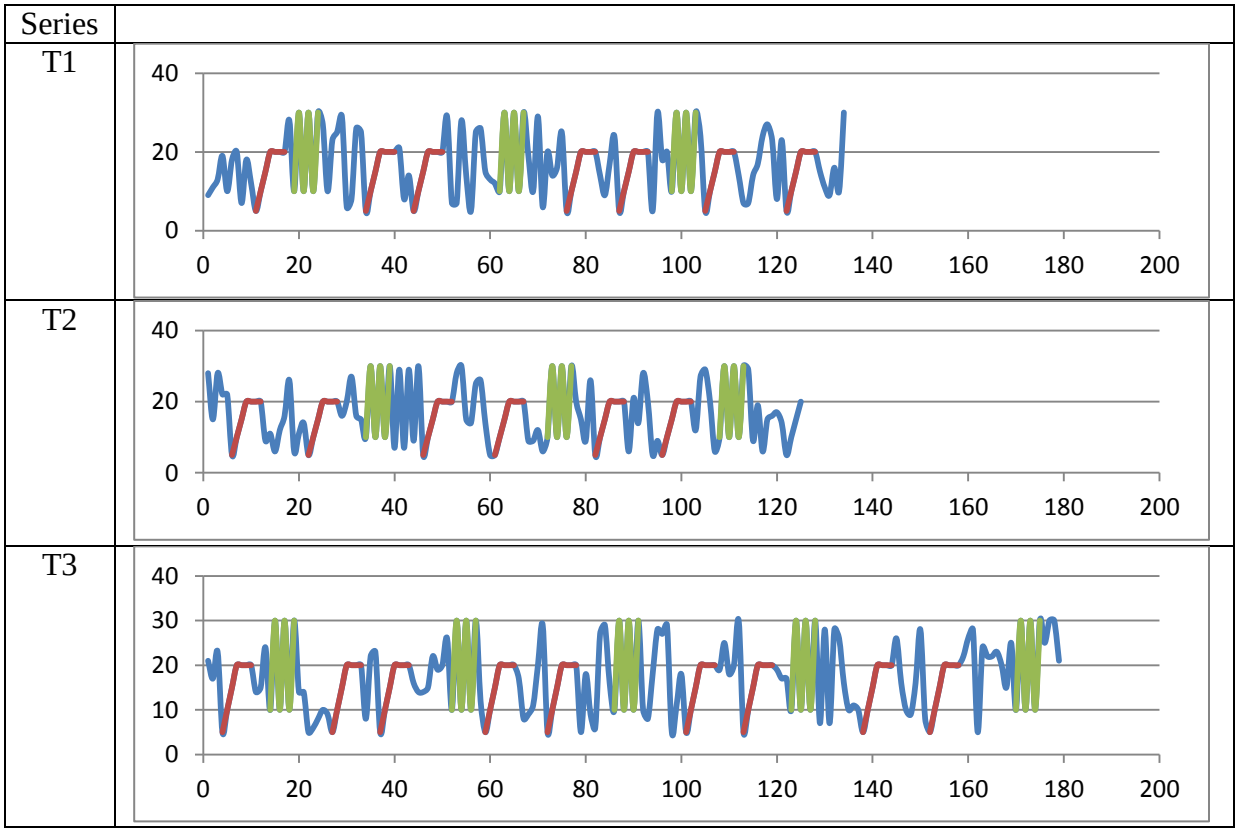


Figure 4.3B - Test set examples of C_1 for Variable 2. Motif A is in red, Motif B in green.

Series	
T1	N M N L N J M J L J L J K K J L N N M M L L L M M J N K L K J J M L J K K J L N J K L L J K K J L N M J J K N N K L N K L M M L L L J L L M N M J N K L J K K J L N N J L N N J L K L J K K J L N J N L M M M L L L K M L J K K J L N N K M K N M J L J J L J K K J L N J K M K M M
T2	L N M L L L J K K J L N N L J K K M M M J L J K K J L N M K M N N M M L L L L N M J K L L L J K K J L N K J L N M J M M L J K K J L N J K JN M M L L L M L J M M L J K K J L N J K L M M K L L J K K J L N J J J M L M M L L L K L J L J N N L L L J K K
T3	J K J L J K K J L N M J N M M L L L L M N M N M K L L J K K J L N N M M L J K K J L N K K J N K N J J M M L L L J J L J K K J L N K M J M K N L J K K J L N K L K J K L N M M L L L M L M L M M K L J K L J K K J L N L K N J K L J K K J L N L K N M M L L L J M M N K M K K K L L J K K J L N N K J N M M N L J K K J L N M M L N L N L J N M N M M L L L L J J K L

Figure 4.3C - Test set examples of C_1 for Variable 3. Motif A is in red, Motif B in green.

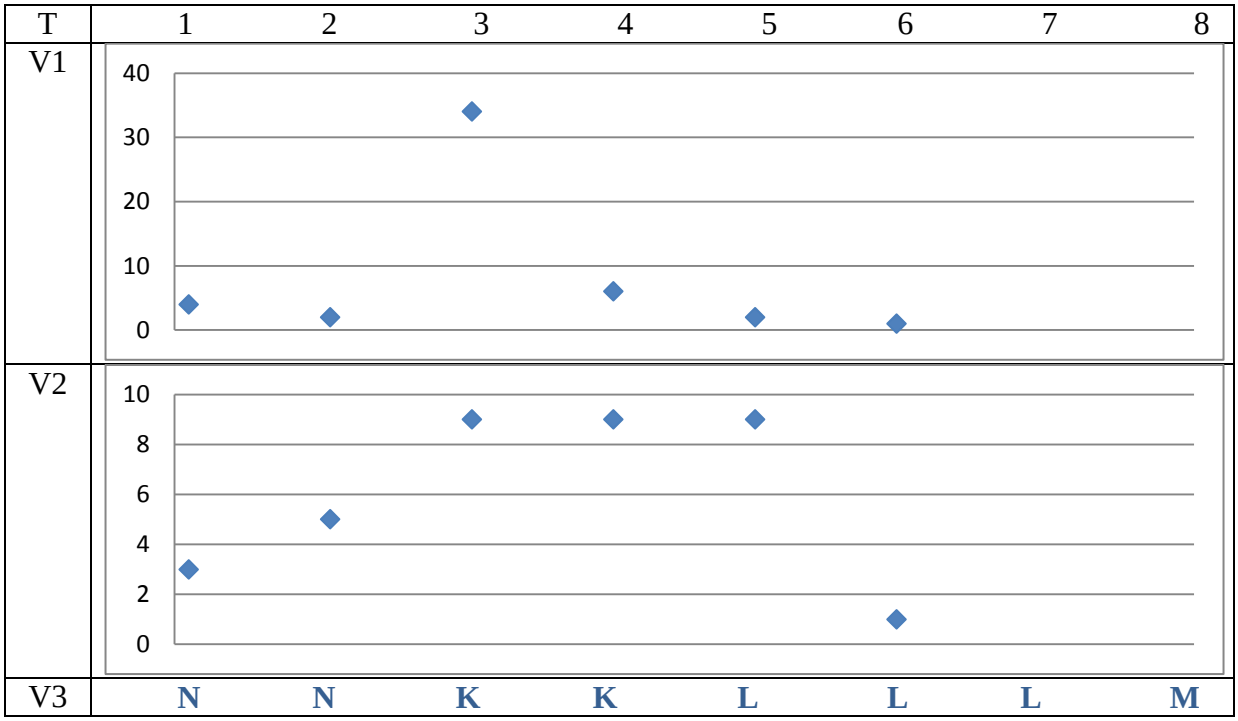


Figure 4.4 - Intermittent BU Motif

Each test set time series has a random length chosen between 2 to 4 years of weekly data. Noise is used to fill between each appearance of a motif using random draws. Each of these noise periods has length uniformly randomly distributed between 0 and 12 with values based on a uniform distribution between the minimum and maximum values of the motifs represented in that time series classification and variable. Examples of the difference in appearance of the time series in the same class are given in Figures 4.3A-C.

A set of business-user-defined motifs is also included in this test case, to demonstrate the utility of the Phase 0. This set will include Motif A, shown in Figure 4.2A, as well as a random length-6 motif displayed in Figure 4.4, which does not have intentional occurrence in the time series.

4.6.2 Approach

4 different clustering runs occur to demonstrate the use of the approach. They are:

1. Using the full window of 6 to 8 on the 3-dimensional time series.
2. Using a partial window of 6 to7 on the 3-dimensional time series.
3. Univariate clustering runs on each of the univariate time series, using the full motif window, [6, 8]. A feature space for Phase 4 is created from the compilation of the individual subsequence clusters.
4. Clustering runs on the individual time series separately without stacking

In all four clustering runs a level of resolution of b=8 is used. The third clustering run represents the concept of "stacking", which has common use in the subsequence-based clustering community to address the issue of multivariate time series with univariate approaches. A Rand index is used to evaluate the results of the approach on the data, with a null model Rand index used as a benchmark.

4.7 Results

4.7.1 Correct Motif Window Classification

Using the motif window of [6,8], 6 subsequence clusters were created. To ensure meaningful F like test statistics, a minimum subsequence size of 8 is set, with an alpha level of .4. Motif occurrence rate by time series is given in Table 4.6. Using this feature space a hard k-means clustering method is used producing the time series clustering also given in Table 4.6.

Table 4.6 - Relative Motif Occurrence and Predicted Class for the Complete Window Case

Time Series	Motifs						Predicted Class
	1	2	3	4	5	6	
T01	1	.	.	.	.	.	1
T02	1	1	1	.	.	.	1
T03	1	2	.	.	.	.	1
T04	.	1	1	.	.	.	1
T05	1	1	.	.	.	.	1
T06	1	.	1	.	.	.	1
T07	1	2	1	.	.	.	1
T08	3	.	.	.	.	.	1
T09	.	.	.	.	.	7	2
T10	.	.	.	.	.	6	2
T11	.	.	.	.	.	8	2
T12	.	.	.	.	.	7	2
T13	.	.	.	.	.	8	2
T14	.	.	.	.	.	5	2
T15	.	.	.	.	.	5	2
T16	.	.	.	.	.	5	2
T17	2	1	.	.	.	3	4
T18	.	.	2	.	.	3	5
T19	1	.	2	.	.	5	5
T20	1	1	2	.	.	5	5
T21	.	.	2	.	.	4	5
T22	1	1	.	.	.	3	4
T23	.	.	1	.	.	3	5
T24	1	.	.	.	.	3	4
T25	.	.	.	1	1	.	3
T26	.	.	.	.	.	.	3
T27	.	.	.	1	.	.	3
T28	.	.	.	2	.	.	3
T29	.	.	.	.	.	.	3
T30	.	.	.	1	1	.	3
T31	.	.	.	3	1	.	3
T32	.	.	.	1	.	.	3

Table 4.7 - Relative Motif Occurrence and Predicted Class for the Incomplete Window Case

Time Series	Motifs							Cluster Predicted
	1	2	3	4	5	6	7	
T01	1	.	.	.	.	.	17	3
T02	1	1	1	.	.	.	13	2
T03	1	2	.	.	.	.	21	2
T04	.	1	1	.	.	.	15	4
T05	1	1	.	.	.	.	12	2
T06	1	.	1	.	.	.	20	4
T07	1	2	1	.	.	.	22	2
T08	3	.	.	.	.	.	25	3
T09	.	.	.	1	.	.	26	1
T10	.	.	.	.	.	.	23	1
T11	.	.	.	.	.	.	28	1
T12	.	.	.	1	.	.	24	1
T13	.	.	.	.	.	.	29	1
T14	.	.	.	.	.	.	24	1
T15	.	.	.	.	.	.	22	1
T16	.	.	.	.	1	.	22	1
T17	1	.	.	.	.	.	14	2
T18	.	.	2	.	.	.	18	4
T19	.	.	1	.	.	.	22	4
T20	1	.	1	.	.	.	22	2
T21	.	.	1	.	.	.	16	4
T22	.	1	.	.	.	.	17	5
T23	.	.	1	.	.	.	17	4
T24	1	.	.	.	.	.	13	3
T25	.	.	.	1	.	.	22	1
T26	.	.	.	.	.	.	21	1
T27	.	.	.	.	.	.	21	1
T28	.	.	.	.	.	1	16	5
T29	.	.	.	.	1	.	16	5
T30	.	.	.	.	.	.	23	1
T31	.	.	.	1	.	.	25	1
T32	.	.	.	.	.	.	19	1

As shown in Table 4.6, the time series were nearly exactly classified, with a resultant Rand index of .971 achieved! Comparing to the null model Rand index value of .77, the subsequence-based clustering method provides increased insight into the classification of similar multivariate time series.

4.7.2 Incomplete Motif Window Classification

Using the smaller motif window of [6, 7], partial subsequences are created which do not encapsulate the full structure of each motif. As a result, a larger number of motifs are discovered, with the inability to be merged in Phase 2 Part 2 due to window limitations. 7 subsequence clusters are created in Phases 1-3 with motif occurrence rates given in Table 4.7. The associated time series classification prediction is also provided in Table 4.7. **A** Rand index of .77 is also achieved using this method, demonstrating the loss in predictive power associated with incorrect window sizing.

4.7.3 Univariate Compilation

Treating each variable as a separate time series, clustering is performed using Chapter 2's approach. Phase 3 Case 3 is used on each univariate time series. The F-like test is used as the similarity measure to mimic the approach used in 4.7.1. Compilation of motif occurrences resultant from Phases 0-3 for these runs is used to create a 'stacked' feature space on which Phase 4 is performed.

Table 4.8 displays the 'stacked' motif occurrence feature space as well as the associated predicted classification for each time series. A Rand index of .969 is achieved using this approach.

Table 4.8 - Relative Motif Occurrence and Predicted Class for the Univariate Runs Case

Time Series	Motifs											Cluster Predicted
	1	2	3	4	5	6	7	8	9	10	11	
T01	1	1	.	.	.	1	1	.	.	.	.	1
T02	2	.	1	.	.	1	1	.	.	.	.	1
T03	3	.	3	.	.	1	1	.	.	.	.	3
T04	1	.	2	.	.	.	.	.	.	.	.	1
T05	2	1	2	.	.	.	1	.	.	.	.	1
T06	3	.	4	.	.	3	.	.	.	.	.	3
T07	3	.	1	.	.	4	.	.	.	.	.	3
T08	5	.	2	.	.	2	.	.	.	.	.	3
T09	.	1	1	1	1	1	.	7	7	7	.	4
T10	.	.	.	.	1	2	1	6	6	5	1	4
T11	.	.	1	.	2	.	.	7	8	5	3	4
T12	.	1	.	.	.	2	.	7	7	4	3	4
T13	.	.	.	.	1	3	2	8	8	6	2	4
T14	.	.	1	.	.	2	.	5	5	5	.	4
T15	.	.	1	.	.	2	1	5	5	3	2	4
T16	.	1	1	1	.	2	.	5	5	3	2	4
T17	7	1	.	.	.	.	.	3	3	2	1	5
T18	5	.	.	.	.	.	.	3	3	2	1	5
T19	7	.	.	.	.	.	.	5	5	4	1	5
T20	9	1	.	.	.	.	.	5	5	4	1	5
T21	6	.	.	.	.	.	.	4	4	3	1	5
T22	7	.	.	.	.	.	.	3	3	2	1	5
T23	5	1	.	.	.	.	.	3	3	3	.	5
T24	3	.	.	.	.	.	.	3	3	2	1	5
T25	.	1	2	2	2	1	1	5	.	4	1	2
T26	.	.	2	.	.	2	.	4	.	3	1	2
T27	.	.	1	1	.	1	1	3	.	2	1	2
T28	.	1	1	1	.	.	.	3	.	2	1	2
T29	.	.	3	.	1	2	.	3	.	3	.	2
T30	.	1	2	.	.	.	2	3	.	1	2	2
T31	.	.	2	.	.	2	.	5	.	3	2	2
T32	.	2	1	1	.	.	.	2	.	2	.	2

Table 4.9 - Motif Size and Origin from Univariate Compilation

Motif	Origin of Motif	Number of Members
1	1	69
2	2	13
3	3	34
4	1	7
5	2	8
6	3	34
7	3	12
8	1	107
9	2	80
10	3	80
11	3	28

4.7.4 Isolated Univariate Approach

Less accurate clustering results occur from analysis on variables individually. Using the three motifs resultant from Phases 1 to 3 on Variable 1, clusters were predicted as given in Table 4.10. This clustering resulted in a rand index of .826.

The 3 motifs created from Variable 2's analysis produced slightly worse results, with a Rand index of .818. The associated clusters are given in Table 4.11. Clustering on Variable 3 produced 5 motifs, but these extra patterns did not correspond to increased classification ability. A Rand index of .729 was achieved with these 5 motifs, cataloged in Table 4.12.

Table 4.10 - Time Series Clustering Using only Variable 1 Motifs

Time Series	Motifs			Cluster Predicted
	1	2	3	
T01	1	.	.	1
T02	2	.	.	4
T03	3	.	.	4
T04	1	.	.	1
T05	2	.	.	4
T06	3	.	.	4
T07	3	.	.	4
T08	5	.	.	4
T09	.	1	7	3
T10	.	.	6	3
T11	.	.	7	3
T12	.	.	7	3
T13	.	.	8	3
T14	.	.	5	3
T15	.	.	5	3
T16	.	1	5	3
T17	7	.	3	2
T18	5	.	3	5
T19	7	.	5	5
T20	9	.	5	2
T21	6	.	4	2
T22	7	.	3	2
T23	5	.	3	5
T24	3	.	3	5
T25	.	2	5	3
T26	.	.	4	3
T27	.	1	3	3
T28	.	1	3	3
T29	.	.	3	3
T30	.	.	3	3
T31	.	.	5	3
T32	.	1	2	3

Table 4.11 - Time Series Clustering Using only Variable 2 Motifs

Time Series	Motifs			Cluster Predicted
	1	2	3	
T01	1	.	.	5
T02	.	.	.	2
T03	.	.	.	2
T04	.	.	.	2
T05	1	.	.	5
T06	.	.	.	2
T07	.	.	.	2
T08	.	.	.	2
T09	1	1	7	1
T10	.	1	6	1
T11	.	2	8	1
T12	1	.	7	1
T13	.	1	8	1
T14	.	.	5	3
T15	.	.	5	3
T16	1	.	5	3
T17	1	.	3	3
T18	.	.	3	3
T19	.	.	5	3
T20	1	.	5	3
T21	.	.	4	3
T22	.	.	3	3
T23	1	.	3	3
T24	.	.	3	3
T25	1	2	.	4
T26	.	.	.	2
T27	.	.	.	2
T28	1	.	.	5
T29	.	1	.	4
T30	1	.	.	5
T31	.	.	.	2
T32	2	.	.	5

Table 4.12 - Time Series Clustering Using only Variable 3 Motifs

Time Series	Motifs					Cluster Predicted
	1	2	3	4	5	
T01	.	1	1	.	.	1
T02	1	1	1	.	.	1
T03	3	1	1	.	.	3
T04	2	.	.	.	.	1
T05	2	.	1	.	.	1
T06	4	3	.	.	.	3
T07	1	4	.	.	.	1
T08	2	2	.	.	.	1
T09	1	1	.	7	.	4
T10	.	2	1	5	1	4
T11	1	.	.	5	3	2
T12	.	2	.	4	3	2
T13	.	3	2	6	2	4
T14	1	2	.	5	.	4
T15	1	2	1	3	2	2
T16	1	2	.	3	2	2
T17	.	.	.	2	1	2
T18	.	.	.	2	1	2
T19	.	.	.	4	1	2
T20	.	.	.	4	1	2
T21	.	.	.	3	1	2
T22	.	.	.	2	1	2
T23	.	.	.	3	.	4
T24	.	.	.	2	1	2
T25	2	1	1	4	1	4
T26	2	2	.	3	1	5
T27	1	1	1	2	1	2
T28	1	.	.	2	1	2
T29	3	2	.	3	.	5
T30	2	.	2	1	2	1
T31	2	2	.	3	2	2
T32	1	.	.	2	.	2

4.7.5 Overall Test Set Results

The correct window approach demonstrates the high level of predictive ability which a multivariate subsequence-based time series clustering approach can provide. A summary of the approaches used and the resultant Rand indexes are given in Table 4.13. The multivariate full update approach of this chapter produces results which are on par with the accuracy of univariate compilation. The univariate compilation run did require additional time for completion of runs, due subsequence eligibility updates and Phase 4 clustering occurring 3 times versus the single run required for Chapter 4's method.

Table 4.13 - Rand index values for each approach

Approach	Rand Index
Multivariate, Correct Window	0.97
Multivariate, Incomplete Window	0.77
Univariate Compilation	0.97
Univariate, Variable 1 only	0.83
Univariate, Variable 2 only	0.82
Univariate, Variable 3 only	0.73

4.8 Conclusions

The multivariate subsequence-based time series clustering approach is a powerful classification tool. Usage of the block motif structure in the motif discovery phases of the approach provide a computationally feasible approach to multidimensional clustering. Even in the test case of intermittent motifs occurring in the time series, the block motif full update approach is shown as useful, resulting in a Rand Index value of .97!

Combining the block motif full update with an intermittent motif discovery and usage in the case of incremental updates will provide a low cost, high clarity understanding of repeated patterns occurring within time series. Examples of the effectiveness of this approach applied to real world data are given in Chapter 5.

CHAPTER 5: MULTIVARIATE CASE STUDIES

5.1 Introduction

The multivariate MDL/Stochastic algorithm presented in Chapter 4 expands on the univariate approach seen in Chapter 2. The multivariate approach can provide insight to a diverse set of multivariate time series, allowing for the interplay between variables in repeated patterns. Case studies on grocer and billiards data highlight the usefulness of this approach, as well as highlight limitations associated with multivariate clustering.

5.2 Grocery Study

5.2.1 Background

Dominick's grocery data is revisited in this section, using the additional capabilities of the multivariate clustering approach. In Chapter 3, sales quantity for individual products is examined to determine any subsequence patterns. The motifs discovered during this study were representative of common retail sub-processes, such as promotions, markdowns, and ramp up periods. Additional structure for sub-processes can be found using multivariate analyses.

Mixed marketing effects of products are often set by the vendor, or parent company. Pricing and promotion decisions are applied weekly to products, resulting in sales unit responses. These strategies are created for individual products, or can be created for products at a higher level, such as product line or parent company. Creating these higher level strategies allows for interaction effects such as cannibalism.

Chapter 3's assertion that these mixed marketing strategies varied by category is reexamined using the added capability of multivariate clustering.

Assertion: Categories have distinctive marketing and pricing strategies.

This assertion's validity will be evaluated using two approaches. The first approach will examine paired revenue streams of top SKUs in a product line to determine if there are distinct marketing and pricing strategies by category. The second approach looks at the mixed marketing effects directly, examining price and promotion type by product.

5.2.2 Approach

The assertion that categories have distinct marketing and pricing strategies is examined in this example, using two test categories: beer and soft drinks. The top 6 revenue grossing brands in each category are used for these analyses. Controlling for other explanatory factors, 24 counts of 12 ounce cans (cases) are used for the SKUs chosen.

In the first approach, two-dimensional sales data consisting of regular and diet/lite average revenue per store is created for each product line. When a product line has multiple products which satisfy the moniker of regular or diet, the product with the larger revenue stream is chosen for inclusion in this study. Table C.3 in Appendix C provides information on the products chosen for this study. A motif window of [4, 10] is used, as was the window in the univariate grocery studies. A level of resolution b=6 is used.

The second approach clusters similar products based upon repeated patterns of price and promotion. Each time series considers only a single product, restricting the approach to only those which are regular (non-diet/lite). A listing of the products used for this approach is given in Table C.3, in Appendix C. Price is produced via revenue divided by sales to ensure any promotion pricing is included in the time series for that week. Promotion, a class variable, has 3 levels: simple price reduction (SPR), Bonus Buy (BB) in which there is an

additional marketing aspect, and no promotion. There is a 4th level available in the data, which indicates the presence of a coupon deal, but this promotion aspect is not used in the history of either category for the top 6 brands. The average occurrence rate by store/week for each of these realizations is used, creating a 3-dimensional time series in total. As in the previous approach, a b-value of 6 and a motif window of [4, 10] is used.

5.2.3 Results

5.2.3.1 Paired Revenue Time Series

The multivariate approach is applied to the 12 paired revenue time series using the specifications given in Appendix B. Phases 1 to 3 resulted in 4 subsequence clusters with sufficient membership for usage in the feature space for Phase 4's clustering. The associated motifs are given in Figure 5.1. The motif occurrence rates for each time series and the associated cluster prediction is given in Table 5.1.

Table 5.1 - Resultant Clusters and Occurrence Rates by Product

Product Name	Motifs (Relative Occurrence)				Cluster Predicted	Category
	107	129	127	131		
Budweiser	0	0	0	0	2	Beer
Busch	0.0044	0	0	0	2	Beer
Coors	0	0	0	0	2	Beer
Miller	0	0	0.0044	0	2	Beer
Old Style	0	0	0	0	2	Beer
Strohs	0	0	0.0049	0.0049	2	Beer
7-Up	0.0025	0.0050	0.0025	0	1	Soft Drinks
Coca Cola	0.0050	0.0025	0.0050	0	1	Soft Drinks
Dr. Pepper	0.0028	0.0056	0.0141	0	1	Soft Drinks
Pepsi	0.0050	0.0050	0.0025	0	1	Soft Drinks
Mountain Dew	0	0.0039	0.0039	0.0078	2	Soft Drinks
Sprite	0	0.0037	0.0037	0.0112	2	Soft Drinks

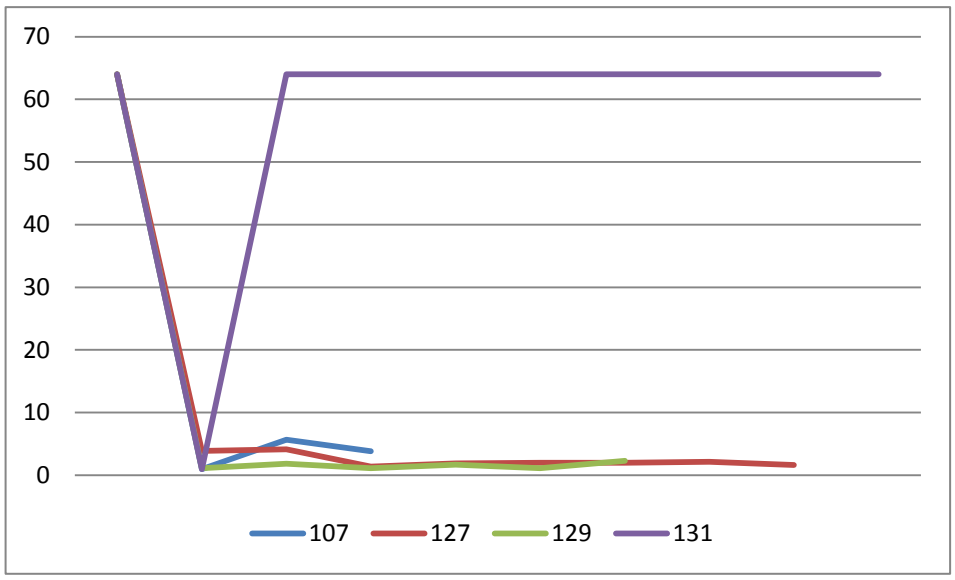
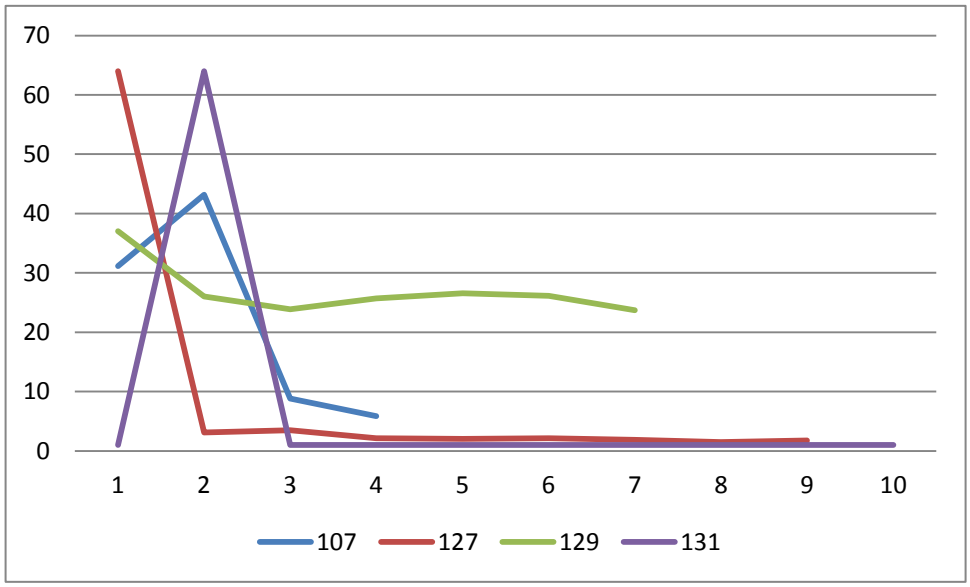
Factor 1	
Factor 2	

Figure 5.1 - Motifs created for use in feature space clustering

These 12 time series are clustered into two classes. Beer was grouped heavily based upon the relative lack of common motifs, while soft drinks had a larger occurrence of motifs including an exclusive presence of Motif 129. Lack of occurrence of motif 107 as well as occurrences of Motif 131 caused Mountain Dew and Sprite to be classified different from the remainder of the soft drink products.

Using the category as the actual classification value, a Rand index of .74 is achieved. Compared to the null model Rand index value of .55, this classification bolsters the assertion that there are distinct sub-processes by category, at least in the cases of soft drinks and beer.

5.2.3.2 Price and Promotion Clustering

Clustering is performed on the pricing and promotion time series for the 12 regular calorie products from the first approach. Eighteen subsequence clusters had sufficient size to be included in the feature space for Phase 4 clustering. The associated motifs are given in Figure 5.2. Table 5.2 displays the occurrence rates and associated clusters for these products. A maximum of 3 clusters were available for Phase 4, for the purpose of allowing an outlier cluster.

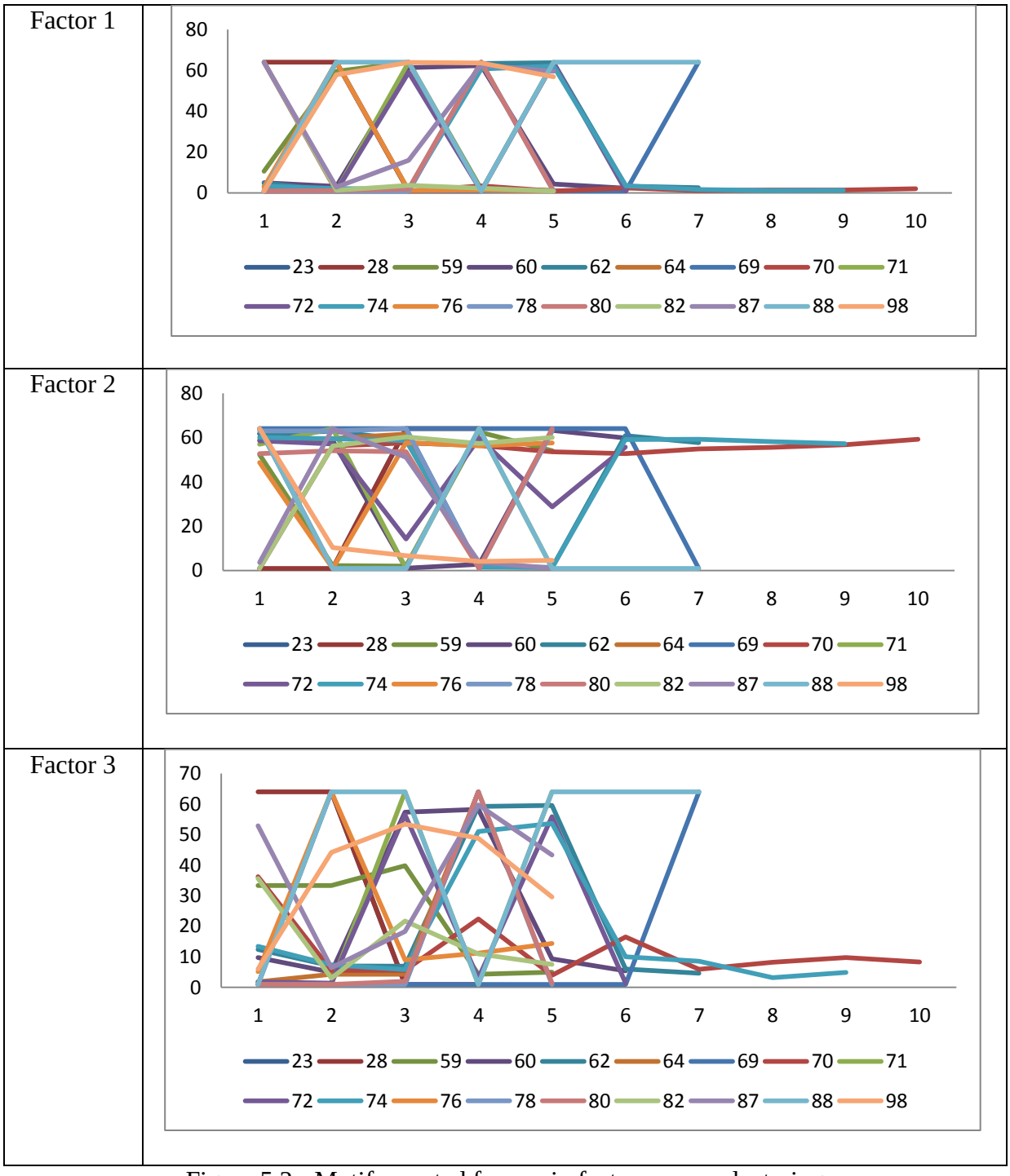


Figure 5.2 - Motifs created for use in feature space clustering

Table 5.2 - Resultant Clusters and Motif Occurrence by Product

Product Name	Motifs																		Cluster	Category
	23	28	59	60	62	64	69	70	71	72	74	76	78	80	82	87	88	98		
Bud	1	12	1	2	2	2	.	1	.	.	.	.	.	.	.	2	.	.	1	Beer
Miller	1	12	.	1	1	1	.	.	.	.	2	1	1	.	.	5	.	.	1	Beer
Old Style	.	10	1	3	1	5	.	2	.	.	.	.	.	.	.	1	.	.	1	Beer
Busch	1	5	1	1	5	4	.	1	.	.	5	2	.	.	1	.	.	.	2	Beer
Coors	.	4	1	2	4	4	.	.	.	.	2	.	.	.	.	4	.	.	2	Beer
Strohs	.	2	.	1	.	3	.	.	1	.	3	.	1	.	1	.	.	.	2	Beer
Coca Cola	1	3	.	1	.	11	.	.	1	2	.	2	6	1	1	.	1	.	3	Soft Drinks
Dr. Pepper	2	1	1	.	1	16	.	2	.	1	.	.	7	.	2	.	.	1	3	Soft Drinks
Mountain Dew	4	1	.	.	1	6	.	1	2	2	.	2	3	.	6	.	1	1	3	Soft Drinks
Pepsi	1	2	.	.	1	13	.	1	1	2	1	1	3	3	3	.	2	.	3	Soft Drinks
7 - Up	2	3	1	.	2	14	1	3	.	1	1	.	7	.	2	.	.	.	3	Soft Drinks
Sprite	1	2	.	1	1	10	1	.	1	1	.	2	5	.	1	.	.	.	3	Soft Drinks

Using the product category as the true clustering of the products, there is an associated Rand index of .88! Comparing with the null approach Rand index value of .55, this approach further bolsters the assertion. Examining motif occurrence, motifs 69, 72, 80, 88, and 98 only occur for soft drinks, and 87 only occurs for beer. There is larger occurrence of major motifs 64 and 82 for soft drinks, and a higher occurrence of motif 28 for beer.

5.2.4 Conclusions-Grocery

Two multivariate approaches to the grocery data are examined in this study. In both cases, the multivariate clustering approach provided category classification higher than expected by the null model. Paired revenue time series provided a predictive ability between categories with a Rand index of .74, mirroring the effectiveness of clustering when using univariate sales curves for identifying categories in Section 3.2.3.1. Use of pricing and promotion information was more telling, with an associated Rand index of .88.

5.3 Billiards Study

5.3.1 Background

Stroke count time series data is studied for the billiard case study in Chapter 3. The goal of this study was to determine whether a class of player, defined by response to pressure, has increased winning percentage when playing against other classes. Clustering upon only stroke count could lead to a partial understanding of responses by players to pressure throughout a match. In this section, two additional variables are examined in an attempt to create a more complete understanding of each player. These variables are chalking and number of shots into an individual's turn.

Chalking consistency is a crucial part of pre-shot routine (Kanov 1999). Lack of chalking prior to a shot increases the likelihood of miscues (a slicing strike upon the cue ball), and can be an indication of a rushed shot. As a result, presence of chalking is a variable of consideration in indirectly identifying response to stress. In contrast to the stress response variables of stroke count and chalking, the potential causal variable of number of shots that a player is into their turn is also examined. Going on a streak can cause an inconsistent player to speed up their shot, or increase their tenseness in the case that they worry about completing the table in a single turn.

Two approaches are considered in this study. The first only includes chalking presence and stroke count to create the multivariate time series. The second approach adds the additional variable of the number of shots a player is into an inning to determine any additional information which can be gained.

5.3.2 Approach

Three variables are recorded from the video recordings of the tournament: strokes per shot, chalking occurrence, and number of shots into the inning. The number of shots into the video segment is used as the time variable.

For the first approach, strokes per shot and chalking presence are used to cluster players. The noisy nature of strokes count (as difficulty of shot not included in analysis) and the binary nature of presence of chalking cause a level of resolution of b=1 to be chosen. An extended motif window of [7, 20] is used, increasing the maximum length from the univariate case study. The second approach adds a third variable of number of shots into the inning that a player is at the time of the current shot. The same level of resolution and motif window will be used.

5.3.3 Results

5.3.3.1 Stroke Count and Chalking

Phases 1-3 applied to the fractured stroke count and chalking time series produced 9 subsequence clusters with positive membership, displayed in Figure 5.3. The motif occurrence rates for each player arecreated after compilation of results across all the relevant time series segments. Table 5.3 provides these relative occurrences as well as the associated player classification.

Table 5.3 - Resultant Clusters and Occurrence Rates by Product

Player ID	Motifs (Relative Occurrence)									Cluster Predicted
	7	9	11	14	15	16	18	20	21	
1	11	0	0	11	0	11	0	0	11	1
13	8	0	0	0	8	0	0	16	16	1
15	14	0	0	0	28	14	0	0	0	1
14	0	0	0	0	0	0	0	0	0	2
2	12	0	19	6	0	0	0	0	0	3
3	7	0	3	3	0	7	0	7	0	3
7	7	0	7	7	0	0	7	0	7	3
8	7	0	11	0	0	4	0	7	7	3
12	6	0	0	0	0	11	0	0	0	3
16	0	0	0	4	4	4	13	0	8	3
4	7	0	0	7	7	7	7	0	0	4
5	0	0	8	0	0	0	8	8	0	4
6	11	0	5	0	5	0	16	0	5	4
9	4	8	0	4	4	0	15	0	0	4
10	6	11	6	0	0	0	17	0	6	4
11	0	0	0	0	0	0	0	0	0	4

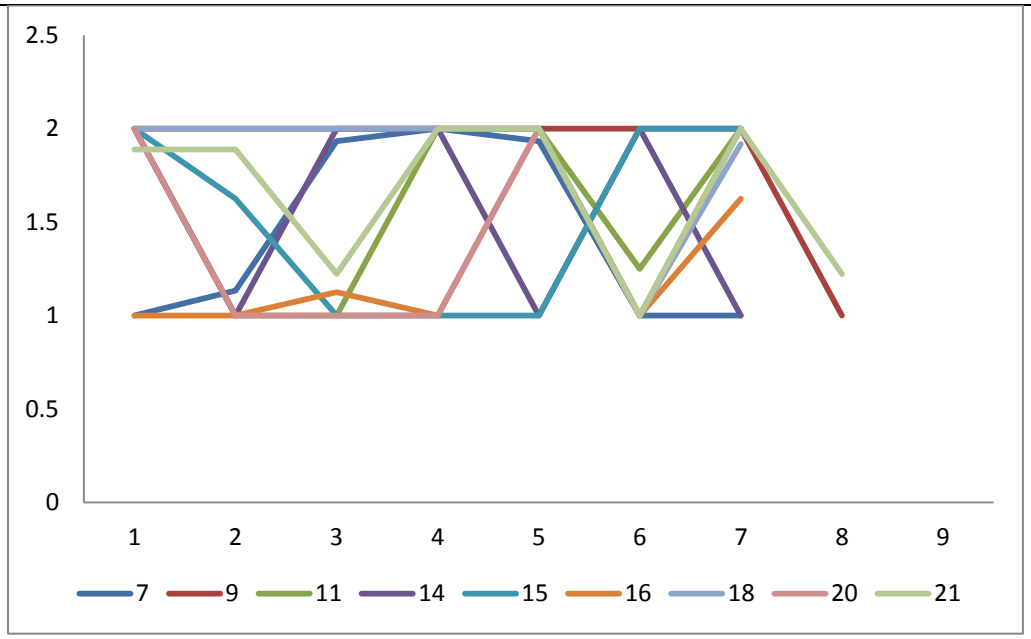
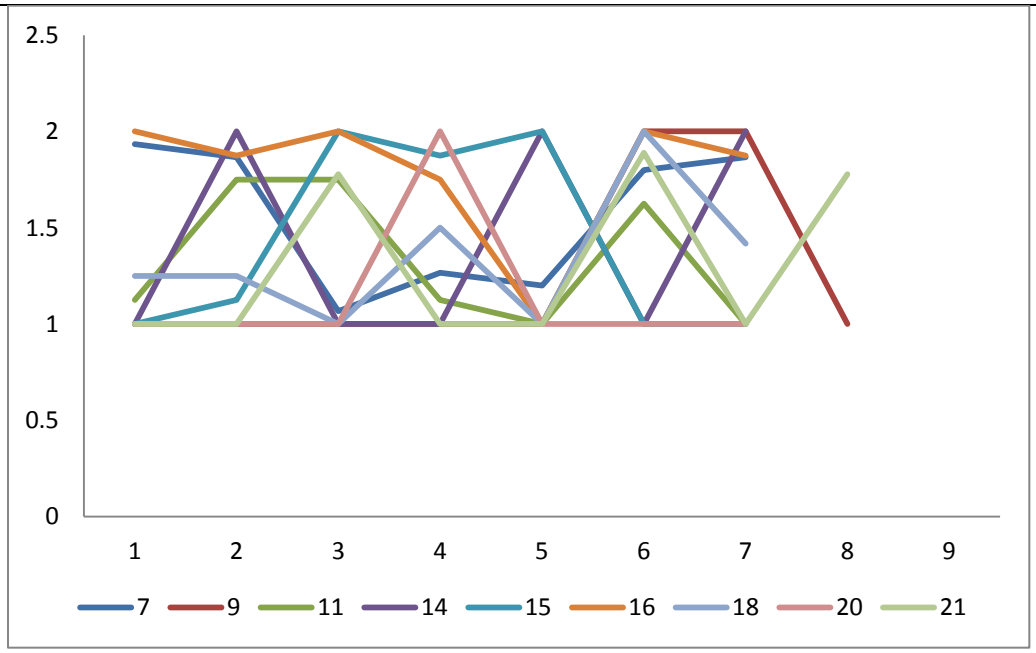
Factor 1	
Factor 2	

Figure 5.3 - Motifs created for use in feature space clustering

The match results of the tournament are given in Appendix E. Using these results, number of wins between classes is determined and given in Table 5.4. Clusters 3 and 4 often beat cluster 1, winning 6 out of 7 matches, including 3 beating 1 3 out of 3 times.

Table 5.4 - Interclass Match Results-Stroke Count and Chalking

Winning Cluster	Losing Cluster	Matches Won	Number Matches
1	2	1	1
1	4	1	4
3	1	3	3
3	4	6	13
4	1	3	4
4	2	1	1
4	3	7	13
3 and 4	1	6	7

Under the assumption that the likelihood for any player from classes 3 or 4 to beat any player from class 1 in a single match is .5 (American Poolplayers Association 2015), the likelihood of at least 6 victories out of 7 occurring for the combined class of 3 and 4 against 1 is .0625. The classification in this example provides results of interest which can be used strategically during league matches. Further experimentation is necessary to gain significance for this result.

5.3.3.2 Stroke Count, Chalking, and Shot in Inning

The addition of the shot into an inning adds an extra dimension to the time series clustering, in an attempt to more fully understand the effects of pressure on a player. This

extra dimension can also add a layer of noise, which can make motif creation difficult. As a result, only two subsequence clusters were created during the process using a minimum subsequence membership of 2. The F-like test for similarity, with an alpha level of .3, further stripped initial results from the bitsave calculation, leaving only 2 observations of each motif.

Table 5.5 - Resultant Clusters and Occurrence Rates by Product

Player ID	Motifs(Relative Occurrence)		Cluster Predicted
	2	3	
1	0	0	1
2	0	0	1
3	0	0	1
4	0	0	1
5	0	0	1
6	0	0	1
7	0	0	1
8	0	0	1
9	0	0	1
10	0	0	1
11	0.01136	0	2
12	0	0	1
13	0	0	1
14	0	0.02817	3
15	0	0	1
16	0	0	1

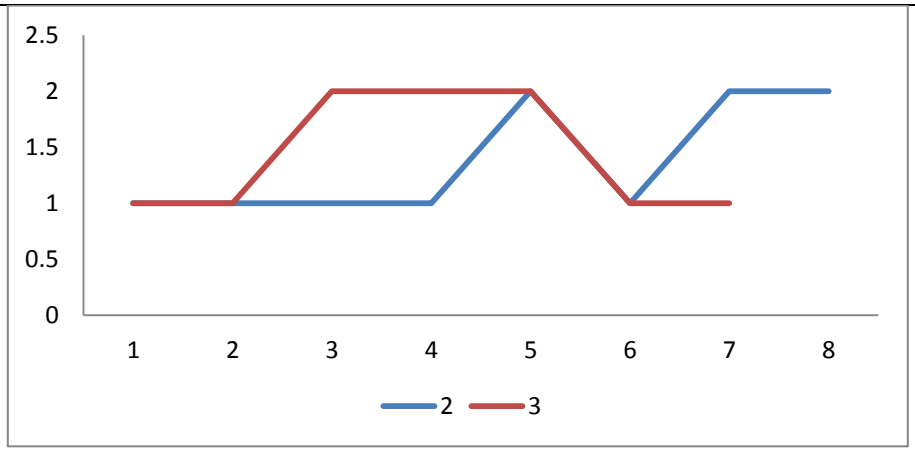
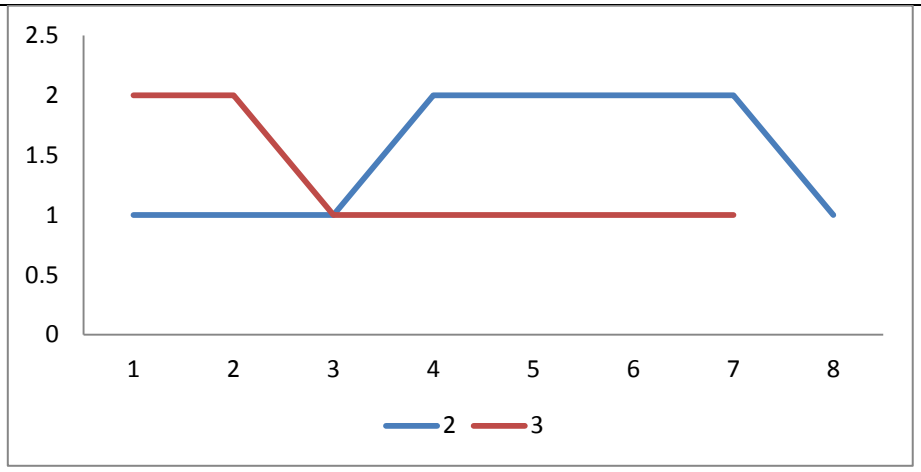
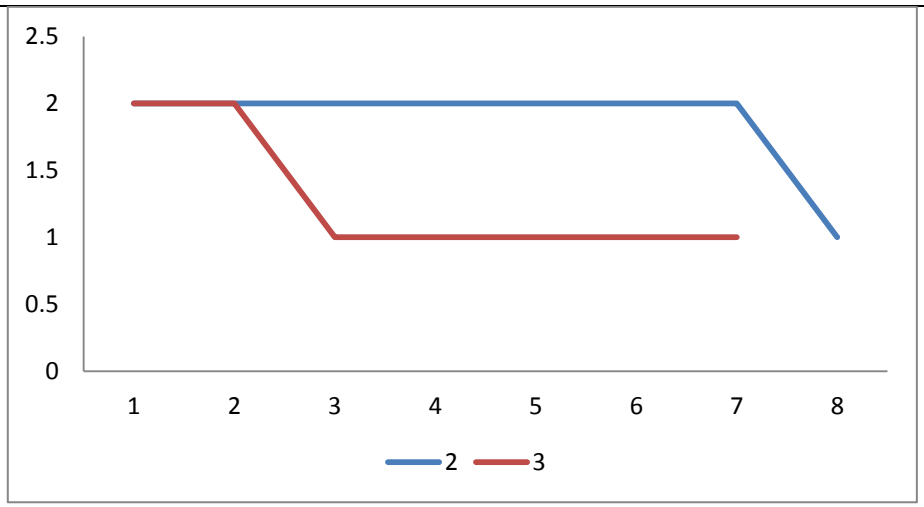
Factor 1	
Factor 2	
Factor 3	

Figure 5.4 - Motifs created for use in feature space clustering

5.3.4 Conclusions-Billiards

The results from the two multivariate billiards examples were mixed. Using the addition of chalking presence to the previously analyzed stroke count produced classes with unexpected interclass winning likelihoods. Using classification merging, the class of players defined by initial classifications 3 and 4 won 6 out of 7 matches against players in class 1. This deviation from expected results can be exploited in strategic choice of player matchups within a team tournament in the future.

The second example did not show the same level of increased insight. The addition of stroke count further muddied the signal in the multivariate time series, resulting in fewer motifs discovered (2 for the second example versus 9 for the first example) and low actionable knowledge with respect to interclass match results.

5.4 Conclusions- Overall

The multivariate approach to subsequence clustering provides an extended framework to the univariate approach outlined in Chapter 2. Allowing the interaction of variables in a multivariate time series dataset, a more comprehensive definition of subsequence motifs is possible than would be had from univariate clustering. The case studies provided in this chapter demonstrate the usefulness of this approach.

Dominick's grocery data is reexamined to find repeated subsequences which are distinctive by category. In both approaches, the classification of products had higher segmentation by category than would be expected from the null model, bolstering the assertion that there are distinctive mixed effects acting upon products within the same category.

The billiards study reexamined the manifestations of pressure on a pool player during a tournament, adding chalking presence to the initial variable of stroke count. The resultant classification of players using this two-dimensional time series provided interclass win results which were outside what was expected given the APA Equalizer® handicapping system. This clustering result provides strategic insight to be used in future tournaments.

The results of the second approach in the billiards study demonstrate a limitation of the multivariate approach in Chapter 4. Block motifs are susceptible to poor clustering in the presence of irrelevant variables. The addition of the number of shots into an inning that a player was did not provide additional information, resulting in a reduction of discovered motifs. This concern as well as other limitations and opportunities for future research are provided in Chapter 6.

CHAPTER 6: LIMITATIONS AND FUTURE RESEARCH

A ‘one-size, fits-all’ approach to time series clustering is an elusive goal. Many clustering methods have been created to address particular time series data concerns, which in turn make them less suitable for alternatively styled data. The approaches given in Chapters 2 and 4 provide a generalized framework for time series clustering using motif discovery and stochastic process estimation in order to produce a quick measure for membership evaluation in the case of incremental data/streaming time series. This approach, like all others, has limitations and concerns which constrain use and require care in implementation. Extensions to this research to address some of these concerns are given in Section 6.2.

6.1 Limitations or Concerns of Work

There are 5 items of concern which are potential limitations to the approach provided in this dissertation.

1. The greediness of the MDL-based motifs discovery approach mixed with motif length restrictions on Action 3 can cause incomplete motifs.
2. The approach has high computational complexity, which can prevent full data motif discovery.
3. Parameter settings can cause significantly different results.
4. Incorrect variable choice in multivariate clustering can lead to reduced insight and poor clustering.

6.1.1-Concerning greedy algorithms

In Phases 1, 2 Part 2, and 3, subsequences are assigned membership to subsequence clusters using bitsave or goodness of fit measures. Only a single action can be performed on any eligible sequence. Upon acquisition of a subsequence by a subsequence cluster, all

subsequences which contain elements of the subsequence(s) chosen are removed from eligibility for further actions.

The greedy approach iterates, such that at each decision stage the best single action is made. This approach's shortness of vision in decision making has the potential to lead to a high number of partial motifs being created. Phase 2 attempts to merge these subsequence clusters together using Action 3. In the case that subsequence clusters have length near the maximum value of the motif window, the lack of available offset lengths for motifs can restrict the usefulness of this phase, as is seen in the example given in Figure 6.1 below.

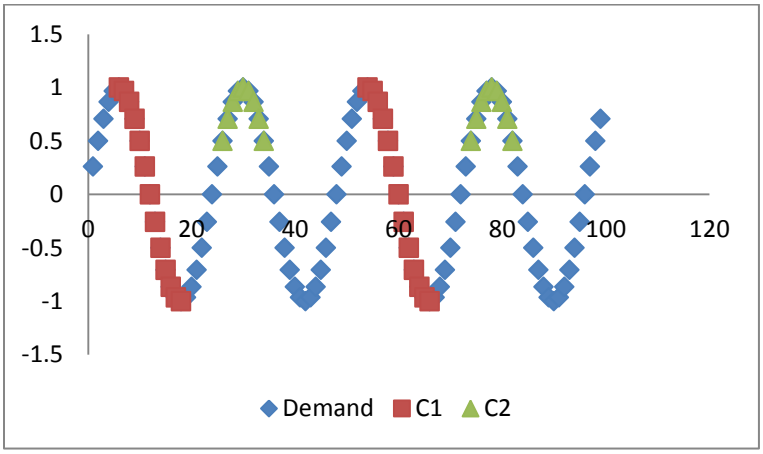


Figure 6.1 - Two motifs with overlapping structure for a time series T

In this example, two subsequence clusters are created for a time series T with a high level of signal, resulting in motifs C_1 and C_2. Subsequences $T_{30,13}$ and $T_{78,13}$ are very similar to motif C_1 but are precluded from potential action due to partial subsequence. Given a motif window which does not allow for subsequence clusters greater than 16 periods, the two

subsequences cannot be appropriately merged in Phase 2 Part 1, resulting in the same subsequence membership at the end of Phase 2.

This restriction on motif merging is especially concerning in the case that there are time series with high signal and low noise. Consider time series T2 in Figure 6.2.

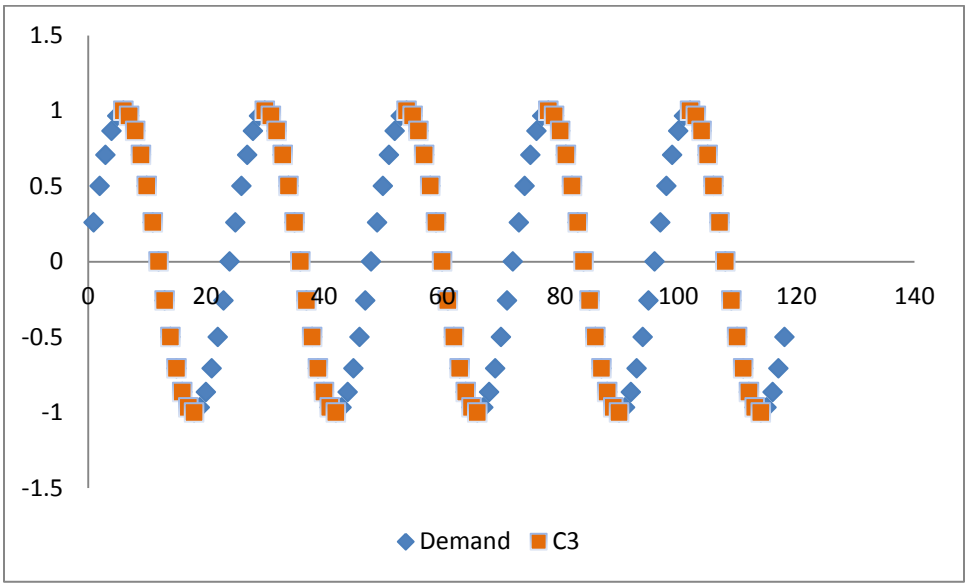


Figure 6.2 - Time Series T2

T2 has a similar series pattern to T, but has a subsequence membership to the cluster with motif C3. This difference in motif occurrence can lead to misclassification of series T as different from time series T2. Subsequence-based time series clustering using MDL has the assumption that there is signal separated by periods of noise. In continuous cyclical signal time series such as T and T2, as well as was seen in the Energy example in Chapter 3, there are no periods of noise.

One approach to address this concern is to artificially fracture the time series on a hypothesized length of the repeating cyclical pattern, increasing the motif window to nearly the entire length of a time series. Upon completion of Phases 1-3, the fractured time series is then recombined, with associated motifs used for the feature space in Phase 4. This approach is utilized for the energy study in Chapter 3 with good results, but relies on a hypothesized fracture length, and restricts the motif window. A second approach to solving this problem is presented in the future research section of this chapter.

6.1.2 Concerning Complexity

The subsequence-based approach to time series clustering offers usage to a wide variety of data, but at a heavy computational cost. Analyses of each of the subsequences associated with a time series increases the number of computations greatly even in the case of bag-of-words approaches, and motif discovery is computationally even more expensive (Rakthanmanon et al. 2012). The full approach outlined in Chapters 2 and 4 focuses on initial rigor at high cost for motif discovery, but providing a low-cost statistical framework to increase speed of incremental-load motif identification. The incremental cost associated with subsequence analysis does not differ significantly from that of a full time series clustering approach.

In order to combat the high levels of effort required for the full approach, restrictive motif window settings can reduce the number of subsequences to be evaluated. Iteration caps are set for Phase 1 and Phase 2 Part 2 to restrict the run time for the algorithm. If necessary, random subsets of overall time series can be used to determine sets of potential

motifs, with the remainder of the time series being evaluated based on the incremental-load approach.

6.1.3 Concerning Parameterization

The approach given in this dissertation requires multiple parameters to be set in order for effective clustering to occur. Incorrect settings for these parameters can lead to unrepresentative motifs and as a result poor classification. The normalization level of resolution, b, as well as the motif window must be set at the beginning of the approach. Iteration maximums for Phases 1 and 2 Part 2 are also set initials, but can be removed to allow for complete results, at the expense of computation time.

Parameters for use in Phase 3 are thresholds. Minimum subsequence cluster size and alpha level thresholds for assessing subsequence membership using chi-square like or F-like tests need to be set. These values can be set after consideration of the results in the first two Phases and are a topic of future research.

Phase 4 requires only choice of clustering method and maximum number of clusters. Multiple criteria for evaluation of clustering such as AIC and Xie-Beni can be used to evaluate the appropriateness of the results of clustering, leading to a parameter-free Phase 4, discussed in Section 6.2.

6.1.4 Concerning Poor Variable Selection

In the multivariate billiards study, player's response to pressure manifested itself in the interplay between chalking and stroke count. Interclass match results showed high inequality in win percentages. This inequality can have great use in future tournaments. The

additional variable of the number of shots into an inning caused the clustering to become poor, with fewer motifs and the loss of these useful interclass match results.

This billiards study illustrates the sensitivity of the multidimensional method to addition of variables which do not have repeated patterns in tandem with the other variables of the time series. Intermittent motifs address this issue, but suffer from high computational effort. Univariate compilation as used in Section 4.6 can also address this issue, but does not address any interplay between the variables.

6.2 Future Research

The approach discussed in this dissertation represents a robust time series classification method which has demonstrated effectiveness on a varied set of examples and applications. A number of future research topics will assist in producing an even more computationally efficient, user-friendly, and effective clustering approach.

Some additions can be created to decrease the number of calculations associated with motif discovery. Given Action 2 on subsequence S and cluster C is chosen as having the best positive bitsave in Phase 1, any other subsequences have a Euclidean distance of 0 with S can be found with minimal additional effort and memory. A maximum size non-overlapping subset of these subsequences can be chosen using a greedy algorithm, with all associated subsequences added to the subsequence cluster C. This can turn the computational effort required for a single iteration into the effect of multiple iterations. Given Phase1 and Phase 2 Part 1 are the high-computational-cost phases of this approach, this reduction could lead to significant reduction in computational cost, especially in the case of low level of resolution b values.

User friendliness is realized in different ways based upon the user. An analyst may prefer a highly parameterized approach to customize the clustering algorithm to best suit the needs of the data, while a business user may request an approach with minimal parameter adjustments. For either user, additional research on determining appropriate parameter settings can be useful. Studies on volatility using a moving average across a time series may provide some knowledge as to the relative size of motifs in which signal is occurring as well as appropriate level of resolution (assuming that the motifs are not volatile). Analysis of the distribution of subsequence cluster sizes at the end of Phase 2 Part 2 can assist in selection of the minimum size threshold. Use of an evaluation criterion in Phase 4 can remove the need to set the number of clusters.

The issue of motif merging and overlay of multiple motifs on a single time series point is a point of great interest in further research. Creation of candidate motifs in Phase 2 Part 1 larger than the original motif window size allows an artificially low initial window size to be used. The restrictive motif window in Phase 1 can reduce computational complexity in this phase, allowing for full motifs to be created in Phase 2 Part 2 and Phase 3.

Another topic of future research is to relax the greedy motif discovery exclusionary rule for subsequences post action. Allowing fuzzy subsequence cluster membership for each time point will allow for overlapping subsequence clusters to not require merging in Phase 2. This approach saves on the complexity of subsequence cluster merging, but adds the additional complexity of fuzzy membership assessment.

Finally, in the case of the multivariate clustering approach, intermittent subsequence clusters can be created after Phase 1 with no additional computational effort. Additional

research into possible greedy approaches to create intermittent motifs during Phase 1, as well as computationally feasible inclusion matrix modification during subsequence cluster updates could potentially produce a more general multivariate clustering method which can accurately create motifs, even in the presence of bad time series variables.

CHAPTER 7: SUMMARY AND CONCLUSIONS

Clustering is common practice in analytics. Data mining techniques such as clustering and classification may be used to create attributes and gain understanding of highly complex data streams. As monitor-based measurement data becomes more plentiful **(**Pereira and De Mello, 2014; Zheng et al., 2011**)**, there is an increased need for fast time series clustering processes which require only a small recent set of raw data. Subsequence-based clustering techniques are apt for this purpose. These clustering techniques can be used for attribute creation, early classification of failure occurrence for in-process systems, and countless other applications.

This dissertation addresses four items not extensively researched in the subsequence-based clustering literature. The first is the production of a comprehensive end-to-end clustering methodology which allows for full motif discovery as well as quick subsequence cluster membership updates in the presence of incremental loads/data streaming. Estimation of stochastic processes, and use of goodness of fit measures, provides a novel way for quick membership evaluation in the case of incremental loads. The extension of the univariate notion of motif to a truly multivariate notion, allows for interplay between variables in a time series not captured by univariate compilation methods. Finally artificial fracturing methods allow greedy subsequence-based approaches to be used on a wider variety of data, including cyclical signaled time series data sets in which no noise periods exist.

Chapters 2 and 4 define the univariate and multivariate approaches to subsequence-based time series clustering. These approaches extended the work of previous subsequence-based approaches (Rakthanmanon et al., 2012), adding on a low cost measurement of

goodness of fit to assess membership, and the structure necessary for multivariate clustering. The addition of business-user-defined motifs prior to Phase 1 assists in reduction of computational time, testing of business theories, and increase in user acceptance.

The test set examples in Chapter 2, as well as real world data examples in Chapter 3 demonstrate the effectiveness of the univariate subsequence clustering approach. In the test examples from Chapter 2, high levels of classification accuracy are achieved, with a Rand Index value of .83 created with no noise added to the motifs. In Chapter 3, three data sets were clustered using the univariate approach: Dominick's grocer data, Duquesne Energy consumption data, and billiards. These studies demonstrate the versatility of the approach on three distinct styles of data: relatively noisy data with motifs, discrete data with continuous cyclical signal, and fractured data respectively.

The grocer's data set resulted in motifs which are common in retail practice. Single week sales spikes with no initial ramp-up are indicative of promotions/price reductions. Ramp-ups across the course of multiple weeks signal goods with multiple week promotion periods. Markdowns occur in seasonal products near the end of their selling period. The resultant time series clustering of sales units across five categories resulted in distinctions in the sales patterns associated with beer and with toothpaste, with a perfect classification of beer category products. The billiards study on stroke count resulted in clustering in which inter-cluster winning between clusters 4 and 5 resulted in 100% winning for cluster 4. Duquesne energy data was fractured artificially using the approach discussed in 6.1.1. The resultant motifs displayed distinct differences in weekday versus weekend motifs, with a

noticeable difference in spike at 2 AM. The resultant clustering on day of week was able to correctly identify week days from weekends.

The test set from Chapter 4 as well as the real world data examples of Chapter 5 demonstrate the utility of the multivariate extension of the approach in analysis of time series in which expression of a motif cannot be fully described by only a single variable. In the test set data in 4.7, a .97 Rand index was achieved using the multidimensional approach, which outperformed each univariate approach!

The real world data examples also demonstrate the effectiveness of this approach. Creating brand-based sales profiles for regular and diet versions of their products, beer was compared soft drinks. Classification of the paired sales resulted in a .74 Rand index, as compared to a null model of .55. The promotion and pricing pairings for products proved more informative, with a Rand index of .88!The billiards study, adding the presence of a chalking as a secondary variable, also observed inter-cluster winning discrepancies as was the case in Chapter 3. The use of the additional variable of number of shots into an inning demonstrated the negative effects of uninformative variables on motif discovery.

The time series clustering approach in this dissertation provides a generalizable approach to determining motifs, adding membership of subsequences, and clustering time series. Goodness of fit tests provide statistical backing to the similarity of a subsequence to a stochastic process, as well as providing a quick evaluation of membership for a new subsequence in the case of incremental loading for in-process systems. The extension to multivariate clustering provides a novel approach to multivariate motif definition, allowing

for intermittent motifs in incremental analysis. This approach is a useful tool in the data analyst's arsenal. Thank you for reading.

REFERENCES

Abdel-Khalek, Gouda. "Income and Price Elasticities of Energy Consumption in Egypt: A Time-series Analysis." *Energy Economics* 10, no. 1 (1986): 47-58. Accessed April 24, 2015. www.sciencedirect.com.

Abdel-Khalik, A Rashad, and Kamal M El-Sheshai. "Sales Revenues: Time-Series Properties and Predictions." *Journal of Forecasting* 2, no. 4 (1983): 351-62. Accessed May 3, 2015. proquest.com.

Abdulla_Al_Maruf, A., and Huang Hung-Hsuan. 'Time Series Classification Method Based On Longest Common Subsequence And Textual Approximation'. *International Conference On Digital Information Management*. IEEE, 2012. 130 - 137. Web. 1 Sept. 2014.

Acosta-Mesa, Héctor-Gabriel, Fernando Rechy-Ramírez, Efrén Mezura-Montes, Nicandro Cruz-Ramírez, and Rodolfo Jiménez. "Application of Time Series Discretization Using Evolutionary Programming for Classification of Precancerous Cervical Lesions." *Journal of Biomedical Informatics* 49 (2014): 73-83. Accessed April 1, 2015. http://www.sciencedirect.com.

Alonso, Carlos, Óscar Prieto, Juan José Rodríguez, and Aníbal Bregón. "Multivariate Time Series Classification via Stacking of Univariate Classifiers." *Studies in Computational Intelligence* 126, no. 1 (2008): 135-51. Accessed April 1, 2015. http://link.springer.com.

American Poolplayers Association. "The Equalizer® Handicap System - American Poolplayers Association." American Poolplayers Association. Accessed May 25, 2015. http://www.poolplayers.com/the-equalizer-handicap-system/.

Artigas, Soiram Ernesto Silva, and Silva Artigas. 'Stochastic Modeling of Lightning Occurrence By Nonhomogeneous Poisson Process'. *International Conference On Lightning Protection*. 2012. Web. 24 Apr. 2015.

Baydogan, Mustafa Gokce, George Runger, and Eugene Tuv. "A Bag-of-Features Framework to Classify Time Series." *IEEE Transactions On Pattern Analysis And Machine Intelligence* 35, no. 11 (2013): 2796-2802. Accessed April 25, 2015. ieeexplore.ieee.org.

Caniato, Federico, Matteo Kalchschmidt, Stefano Ronchi, Roberto Verganti, and Giulio Zotteri. "Production Planning & Control: The Management of Operations." *Production Planning & Control* 16, no. 1 (2005): 32-43. Accessed April 24, 2015. http://www.tandfonline.com.

Chan, Franky Kin-Pong, Ada Wai-chee Fu, and Clement Yu. "Haar Wavelets for Efficient Similarity Search of Time-Series: With and Without Time Warping." *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING* 15, no. 3 (2003): 686-705. Accessed April 1, 2015. http://ieeexplore.ieee.org.

Chaovalitwongse, Wanpracha Art, Oleg A. Prokopyev, and Panos M. Pardalos. "Electroencephalogram (EEG) Time Series Classification: Applications in Epilepsy." *Annals of Operations Research* 148, no. 1 (2006): 227-50. Accessed April 19, 2015. search.proquest.com.

Chahrour, Ryan A. "Sales and Price Spikes in Retail Scanner Data." *Economics Letters* 110, no. 2 (2010): 143-46. Accessed May 4, 2015. www.sciencedirect.com.

Chen, Yixin, Jinbo Bi, and J.Z. Wang. "MILES: Multiple-Instance Learning via embedded Instance Selection." IEEE Transactions on Pattern Analysis and Machine Intelligence 28, no. 12 (2006): 1931-1947. Accessed June 10, 2014. ieeexplore.ieee.org.

Chicago, University of. "Dominick's Database." Kilt's Center for Marketing. Accessed May 4, 2015. http://research.chicagobooth.edu/kilts/marketing-databases/dominicks/.

Denton, Anne, Christopher Besemann, and Dietmar Dorr. "Pattern-based time-series subsequence clustering using radial distribution functions." *Knowledge and Information Systems* 18, no. 1 (2009):1-27. Accessed September 9th, 2014. http://link.springer.com.

Dumitrescu, D., Beatrice Lazzerini, and L.C. Jain. *Fuzzy Sets and Their Application to Clustering and Training*. Boca Raton, FL: CRC Press, 2000.

Duquesne Light. "Inside Duquesne Light." Duquesne Light. Accessed May 25, 2015.

Fok, Dennis, Philip Hans Franses, and Richard Paap. "Seasonality and Non-linear Price Effects in Scanner-data-based Market-response Models." *Journal of Econometrics* 138 (2007): 231-51. Accessed May 4, 2015. www.sciencedirect.com.

Fu, Tak-Chung. "A Review on Time Series Data Mining." *Engineering Applications of Artificial Intelligence* 24, no. 1 (2011a): 164-81. Accessed April 1, 2015. www.sciencedirect.com.

Fu, Zhouyu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. "Music Classification via the Bag-of-Features Approach." Pattern Recognition Letters 32, no. 14 (2011b): 1768-1777. Accessed September 9th, 2014. www.sciencedirect.com.

Gallager, Robert. "Gaussian Random Vectors and Processes." In Stochastic Processes: Theory For Application, 109-165. Cambridge Press, 2013.

Guo, Chonghui, Hongfeng Jia, and Na Zhang. "Time Series Clustering Based on ICA for Stock Data Analysis."*Wireless Communications, Networking and Mobile Computing*, 2008, 1-4. Accessed April 1, 2015. http://ieeexplore.ieee.org.

Goulart, Antonio, Rodrigo Guido, and Carlos Maciel. "Exploring Different Approaches for Music Genre Classification." *Egyptian Informatics Journal* 13, no. 1 (2012): 59-63. Accessed October 1, 2014. www.sciencedirect.com.

Gould, Stephen Jay. *Punctuated Equilibrium*. Cambridge, Mass.: Belknap Press of Harvard University Press, 2007.

Hennig, Christian. "Package 'fpc'" R-Project. October 1, 2104. Accessed April 26, 2015. http://cran.r-project.org/web/packages/fpc/fpc.pdf.

Hoell, Simon, and Piotr Omenzetter. "Structural Damage Detection in Wind Turbine Blades Based on Time Series Representations of Dynamic Responses." *Conference: Proceedings of SPIE's 2015 Conference on Smart Structures and Materials/Nondestructive Evaluation and Health Monitoring* 94390B (2015): 1-11. Accessed January 1, 2015. https://www.researchgate.net.

Hong, Tao. "Crystal Ball Lessons in Predictive Analytics." EnergyBiz 12, no. 2 (2015): 35-37.

Jadhav, Swapnil Ashok, D. V. L. N. Somayajulu, Nagesh Bhattu, R.B.V. Subramanyam, and P. Suresh. "Context Dependent Bag of Words Generation." *Advances in Computing, Communications and Informatics*1, no. 1 (2013): 1526-531. Accessed April 1, 2015. http://ieeexplore.ieee.org.

Jeong, Young-Seon, Myong Jeong, and Olufemi Omitaomu. "Weighted Dynamic Time Warping for Time Series Classification." *Pattern Recognition* 44, no. 1 (2011): 2231-240. Accessed June 1, 2014. www.elsevier.com/locate/pr.

Kabacoff, Robert I. "Cluster Analysis." Quick-R. January 1, 2014. Accessed April 26, 2015. http://www.statmethods.net/advstats/cluster.html.

Kanov, Gerry, and Shari Stauch. *Precision Pool*. Champaign, IL: Human Kinetics, 1999.

Keogh, Eamonn. "Fast Similarity Search in the Presence of Longitudinal Scaling in Time Series Databases."*IEEE*, (1997): 578-84. Accessed April 5, 2015. ieexplore.ieee.org.

Keogh, Eamonn, and Shruti Kasetty. "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration." *Data Mining and Knowledge Discovery* 7, no. 4 (2003): 349-71. Accessed September 1, 2014. link.springer.com.

Keogh, Eamonn, and Jessica Lin. "Clustering of time-series subsequences is meaningless: implications for previous and future research." *Knowledge and Information Systems* 8, (2005): 154-177. Accessed September 9, 2014http://link.springer.com.

Košmelj, Katarina, and Vladimir Batagelj. "Cross-sectional Approach for Clustering Time Varying Data."*Journal of Classification* 7, no. 2 (1990): 99-109. Accessed April 1, 2015. http://link.springer.com.

Kumar, Mahesh, and Nitin Patel. "Clustering Data with Measurement Errors." *Computational Statistics and Data Analysis* 51, no. 1 (2007): 6084-101. Accessed April 1, 2015. www.sciencedirect.com.

Lee, Yi-Shian, and Lee-Ing Tong. "Forecasting Nonlinear Time Series of Energy Consumption Using a Hybrid Dynamic Model." *Applied Energy* 94, no. 1 (2012): 251-56. Accessed April 24, 2015. www.sciencedirect.com.

Leemis, Larry. "Estimating and Simulating Nonhomogeneous Poisson Processes." May 23, 2003. Accessed September 9, 2014. http://www.math.wm.edu/~leemis/icrsa03.pdf.

Lewis, P.A.W., and G.S. Shelder. "Simulation of Nonhomogeneous Poisson Processes with Log Linear Rate Function." *Biometrika* 63, no. 3 (1976): 501-05. Accessed September 1, 2014. www.jstor.org.

Lian, Xiang, and Lei Chen. "Efficient Similarity Search over Future Stream Time Series." *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING* 20, no. 1 (2008): 40-54. Accessed April 24, 2015. ieexplore.ieee.org.

Liao, T. Warren. "Clustering of Time Series Data - A Survey." *Pattern Recognition* 38, no. 11 (2005): 857-74. Accessed September 9, 2014. www.sciencedirect.com.

Lin, Jessica, Eamonn Keogh, Li Wei, and Stefano Lonardi. "Experiencing SAX: a Novel Symbolic Representation of Time Series." Data Mining and Knowledge Discovery 15, no. 2 (2007): 107-144. Accessed September 9, 2014. http://link.springer.com.

Massey, William A., Geraldine A. Parker, and Ward Whitt. "Estimating the parameters of a Nonhomogeneous Poisson process with Linear Rate." Telecommunication Systems 5, no. 2 (1996): 361-388. Accessed September 9, 2014. http://link.springer.com.

Miyamoto, Sadaaki, and Hidetomo Ichihashi. *Algorithms for Fuzzy Clustering Methods in C-means Clustering with Applications*. Berlin: Springer, 2008.

Moller-Levet, C.S., F. Klawonn, K.-H. Cho, H. Yin, and O. Wolkenhauer. "Clustering of Unevenly Sampled Gene Expression Time-series Data." *Fuzzy Sets and Systems*, 152, no. 1 (2005): 49-66. Print.

Morrill, Jeffrey. "Distributed Recognition of Patterns in Time Series Data." *Communications of the ACM* 41, no. 5 (1998): 45-51. Accessed April 1, 2015. http://dl.acm.org.

Mueen, Abdullah, Eamonn Keogh, Qiang Zhu, Sydney S. Cash, M. Brandon Westover, and Nima Bigdely-Shamlo. "A Disk-aware Algorithm for Time Series Motif Discovery." *Data Mining Knowledge Discovery* 22 (2010): 73-105. Accessed July 1, 2014. http://link.springer.com.

Nowak, Eric, Frederic Jurie, and Bill Triggs. "Sampling Strategies for Bag-of-Features Image Classification." *Computer Vision* 3954, (2006): 490-503. Accessed July 1, 2014. http://link.springer.com.

Nguyen, Hai-Long, Wee-Keong Ng, and Yew-Kwong Woon. "Closed Motifs for Streaming Time Series Classification." *Knowledge Information Systems* 41, no. 1 (2014): 101-25. Accessed April 25, 2015. http://link.springer.com.

Pedan, Alex. "Analysis of Count Data Using the SAS System." Statistics, Data Analysis, and Data Mining. Accessed September 1, 2014. http://www2.sas.com/proceedings/sugi26/p247-26.pdf.

Pereira, Cassio, and Rodrigo De Mello. "TS-stream: Clustering Time Series on Data Streams." *Journal of Intelligent Information Systems* 42, no. 3 (2014): 531-66. Accessed April 1, 2015.

Qiu, Yingning, Yanhui Feng, Peter Tavner, Paul Richardson, Gabor Erdos, and Bindi Chen. "Wind Turbine SCADA Alarm Analysis for Improving Reliability." *Wind Energy* 15, no. 8 (2012): 951-66. Accessed April 1, 2015. http://onlinelibrary.wiley.com.

Qamar, Ihtzaz. "Method to Determine Optimum Number of Knots for Cubic Splines." *Communications in Numerical Methods in Engineering* 9, no. 6 (1993):483-488. Accessed September 9, 2014. http://onlinelibrary.wiley.com.

Rakthanmanon, Thanawin, Eamonn Keogh, Stefano Lonardi, and Scott Evans. "MDL-based Time Series Clustering." *Knowledge and Information Systems* 33, no. 2 (2012): 371-99. Accessed September 9, 2014. http://link.springer.com.

Rand, William M. "Objective Criteria for the Evaluation of Clustering Methods." *Journal of American Statistical Association* 66, no. 336 (1971): 846-850. Accessed September 9, 2014. www.jstor.org.

Rogers, Simon, and Mark Girolami. *A First Course in Machine Learning*. Boca Raton: CRC Press, 2012.

Ross, Sheldon. "The Exponential Distribution and the Poisson Process." In *Introduction to Probability Models*, 291-352. 10th ed. Academic Press, 2010.

SAS. "SAS/STAT Software - Clustering Analysis." SAS. Accessed April 1, 2015. http://support.sas.com/rnd/app/stat/procedures/ClusterAnalysis.html#cluster.

Sakoe, Hiroaki, and Seibi Chiba. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition." *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-26, no. 1 (1978): 43-49.

Sannino, Giovanna, Ivanoe De Falco, and Giuseppe De Pietro. "Monitoring Obstructive Sleep Apnea by Means of a Real-time Mobile System Based on the Automatic Extraction of Sets of Rules through Differential Evolution." *Journal of Biomedical Informatics* 49 (2014): 73-83. Accessed April 12, 2015. www.sciencedirect.com.

Silva, Johnathan, Elain Faria, Rodrigo Barros, Eduardo Hruschka, Andra Carvalho, and Joao Gama. "Data Stream Clustering: A Survey." *ACM Computing Surveys (CSUR)* 46, no. 1 (2013): 1-31. Accessed April 2, 2015. http://dl.acm.org.

Stack Overflow. "Generate Correlated Data in Python." Stack Overflow. May 25, 2015. Accessed May 25, 2015. http://stackoverflow.com/questions/16024677/generate-correlated-data-in-python-3-3.

Stine, Robert. "Model Selection Using Information Theory and the MDL Principle." *Sociological Methods and Research* 33, no. 2 (2004): 230-60. Accessed May 1, 2014. smr.sagepub.com.

Tipping, Michael E., and Christopher M. Bishop. "Probabilistic Principal Component Analysis." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61, no. 3 (2002): 611-22. Accessed April 30, 2015. onlinelibrary.wiley.com.

Varahan, S.R. Srinivasa. "Poisson Processes." Accessed September 9, 2014. *Http://www.math.nyu.edu/faculty/varadhan/spring06/spring06.1.pdf*.

Wang, Yuedong. *Smoothing Splines: Methods and Applications*. Boca Raton, FL: CRC Press, 2011.

Ye, Lexiang, and Eamonn Keogh. "Time Series Shapelets: A New Primitive For Data Mining." *KDD*, 2009, 947-55. Accessed July 1, 2014. ACM Digitial Library.

Ye, Lexiang, and Eamonn Keogh. "Time Series Shapelets: A Novel Technique That Allows Accurate, Interpretable and Fast Classification." *Data Mining Knowledge Discovery* 22 (2011): 149-82. Accessed July 1, 2014. http://link.springer.com.

Yilmaz, Bülent, Ugur Cunedioglu, and Engin Baysoy. "Usage of Spline Interpolation in Catheter-based Cardiac Mapping." *Turkish Journal of Electrical Engineering & Computer Sciences* 18, no. 6 (2010): 989-1002. Accessed April 11, 2015. http://web.a.ebscohost.com.

Zheng, Jun, David Simplot-Ryl, Chatschik Disdikian, and Hussein T. Mouftah. "The Internet of Things."*Communications Magazine, IEEE* 49, no. 11 (2011): 30-31. Accessed April 1, 2015. ieexplore.ieee.org.

Zhu, Fukang. "Modeling Time Series of Counts with COM-Poisson INGARCH Models." *Mathematical and Computer Modelling* 56, no. 9-10 (2012): 191-203. Accessed April 24, 2015. www.sciencedirect.com.

Zolhavarieh, Seyedjamal, Saeed Aghabozorgi, and Ying Wah Teh. "A Review of Subsequence Time Series Clustering." *The Scientific World Journal* 2014, no. 1 (2014): 1-19. Accessed April 24, 2015. http://www.hindawi.com/journals/tswj/2014/312521/.

APPENDICES

APPENDIX A – PSEUDOCODE

ED(SS_i,SS_j)-Weighted Euclidean Distance (Weighted Upon Eigenvalues)

L=length SS_i
Nvar=Number of variables or factors of SS_i
If Nvar=1
    $Eigen_1$=1
Else
    $Eigen_k$=Eigenvalue associated with factor k
End
Euclid_Dist=0
For V=1 to Nvar
    $Var\_ed = \sqrt{\sum_{p=1}^{L}(SS_i[p] - SS_j[p])^2}$
    Euclid_Dist=Euclid_Dist+$Eigen_V$*Var_ed
End

DL(T) – Description length for subsequence T

L=length of T
Count frequency of realizations R_1 to R_{2^b}
$R_{Total}=\sum R_k$
$P(T=R_k)=R_k/R_{Total}$
$H(T) = -\sum_{k=1}^{R_{2^b}} P(T=t)\log_2 P(T=t)$
//Uses caveat that if $P(T=R_k)=0$ then $P(T=R_k)\log_2 P(T=R_k)=0$
Description Length=L*H(T)

Phase 0-Univariate
Initialize Clusters as Null
Add BU Defined Motifs
For Motif Length= <MM_Min> to <MM_Max>
    For each pairing of subsequences SS_i and SS_j
        $EuclideanDistance_{i,j}$=**ED**(SS_i,SS_j)
    End
End

Phase 0-Multivariate
Normalize numerical values to 0 mean and unit variance
For each nonordinal variable Var_i

NR_i=Number of realizations of Var_i-1
If NRi ≤ <Max_Realizations>
Create dummy variables $X_{i,1},\ldots, X_{i,NRi}$ s.t. $X_{i,k} = \begin{cases} 1, if\ realization\ occurs \\ 0, otherwise \end{cases}$
Else
Find most frequent <Max_Realizations>-1 realizations of Var_i.
For these realizations, create dummy variables $X_{i,1},\ldots, X_{i,<Max\_Realizations>-1}$
Create remaining realization variable $X_{i,<Max\_Realizations>}$
End
Normalize to 0 mean and sum of variances=1
End
Run PCA to create factors for input data set.
Create associated eigen values for each factor
Create PCA transformed versions of business user defined motifs.
For Motif Length= <MM_Min> to <MM_Max>
For each pairing of subsequences SS_i and SS_j
$EuclideanDistance_{i,j}$=**ED**(SS_i,SS_j)
End
End

Phase 1
Initialize update to 1, initialize iteration counter to 1, set maximum number of iterations
Do While Iteration< Cutoff for iterations and Update=1
For Subsequence Length L=<MM_Min> to <MM_Max>
//Action 1
BESTSS2SS=(SS_1^*,SS_2*)|**ED**(SS_1^*,SS_2*) ≤ **ED**(SS_i,SS_j) for all SS_i, SS_j
C_L=mean(SS_1^*,SS_2*)
$Diff_{1,L}$=SS_1-C_L
$Diff_{2,L}$=SS_2-C_L
Cost new=**DL**($Diff_{1,L}$)+**DL**($Diff_{2,L}$)+**DL**(C_L)
Cost old=**DL**(SS_1)+**DL**(SS_2)
$Bitsave_{S2S,L}$=Cost old-Cost new
//Action 2
For each motif C_k of length L
BestSS$_k$=SS* s.t. **ED**(C_k, SS*)≤ **ED**(C_k,SS_i) for all SS_i
M={$SS_{i,k}$ | $SS_{i,k}$ has membership with motif C_k}
Index={i | $SS_{i,k}$ has membership with motif C_k}
C_k*=mean(SS*, $SS_{i,k}$∈M)
Diff*=C_k*-SS*
Do for $SS_{i,k}$∈M
$Diff_{i,k*} = SS_{i,k} - C_{k*}$
End

$$Cost\ Old = \sum_{i \epsilon Index} \boldsymbol{DL}(Diff_{i,k}) + \boldsymbol{DL}(C_k) + \boldsymbol{DL}(SS^*)$$

$$Cost\ New = \sum_{i \epsilon Index} \boldsymbol{DL}(Diff_{i,k^*}) + \boldsymbol{DL}(C_k{}^*) + \boldsymbol{DL}(Diff^*)$$

Bitsave$_{C2S}$(k)=Cost Old-Cost New

End

End

$Best\ Bitsave = \min(\min_k \text{BitsaveC2S(k)}, \min_L \text{Bitsave}_{\text{S2S,L}})$

If Best Bitsave>0

If Best Bitsave a S2S(SS$_1$ and SS$_2$ the two subsequences)

Save new $C_L = mean(SS_1, SS_2)$ in motif list

Remove SS$_1$, SS$_2$ and any subsequences which overlap with SS$_1$ or SS$_2$ from consideration for future actions.

End

IF Best Bitsave a C2S(C_k the original motif, SS* the Subsequence)

$C_k = C_{k*}$

Remove SS* and any Subsequence which overlap with SS* from consideration for future actions.

End

End

Iteration=Iteration+1

End

Phase 2-Part 1

Let Update=1

Set all motifs to active status. Num_Motif$_r$=1 for all motifs C$_r$.

Do while update=1

X=number of active motifs

Do s=1 to X

Do t=s to X

If C$_s$ and C$_t$ have not been compared before

For each motif overlap with total resultant length≤**MM_Max**

Let C*=Specified overlay of C$_s$ and C$_t$

C$_s$*=C$_s$ portion of overlay

C$_t$*=C$_t$ portion of overlay

M$_s$={SS$_{i,s}$ | SS$_{i,s}$ has membership with motif C$_s$}

Index$_s$={i | SS$_{i,s}$ has membership with motif C$_s$}

M$_t$={SS$_{j,t}$ | SS$_{j,t}$ has membership with motif C$_t$}

Index$_t$={j | SS$_{j,t}$ has membership with motif C$_t$}

Cost old=$\sum_{\substack{i \epsilon Index_s \\ j \epsilon Index_t}} \boldsymbol{DL}(Diff_{i,s}) + \boldsymbol{DL}(Diff_{j,t}) +$

$\boldsymbol{DL}(C_s) + \boldsymbol{DL}(C_t)$

$$Cost\ New = \sum_{\substack{i \epsilon Index_S \\ j \epsilon Index_t}} \boldsymbol{DL}(Diff_{i,s*}) + \boldsymbol{DL}(Diff_{j,t*}) + \boldsymbol{DL}(C^*)$$

Bitsave(overlay,s,t)=Cost Old-Cost New

End

Bitsave(s,t)=max(Bitsave(overlay,s,t))

End

End

End

$Best\ Bitsave = \max_{s,t \in X} \text{Bitsave}(s,t)$

If Best Bitsave>0

//Candidate motif could be a fuller image of a sub-process

Let s and t be the associated moitfs to the best bitsave, and C* the associated best merged motif (highest bitsave)

Add C* to list of motifs and set as active.

$Num\_Motif_* = Num\_Motif_s + Num\_Motif_t$

Set C_s and C_t to inactive status.

End

End

Phase 2-Part 2

Max_Motif_Merger=max(Num_Motif)

For N=Max_Motif_Merger to 1 by -1 steps

Iter=1

Update=1

Do while Iter<Max_Clusters and Update=1

Let R_N be the set of motifs which have Num_Motif=N

Let $X_N=|R_N|$

For k=1 to X_N

//Cluster C_k in R_N

BestSS$_k$=SS* s.t. **ED**(C_k, SS*)≤ **ED**(C_k,SS$_i$) for all SS$_i$

Diff=C_k-SS*

$Cost\ Old = \boldsymbol{DL}(SS^*)$

$Cost\ New = \boldsymbol{DL}(Diff^*)$

$Bitsave_{C2S}(k)$=Cost Old-Cost New

End

Find maximum bitsave MAXBIT

If MAXBIT>0

Add membership of SS* associated with MAXBIT to C_k

Update=1

```
                End
                If MAXBIT<=0
                        Update=0
                End
                Iter=Iter+1
        End
End
```

Phase 3Case 1(Poisson Approach)

Remove motfs with membership size < MIN_SIZE_P3>
Let M be the set of motifs remaining
For C_k in M
- Determine number of knots(nodes) for cubic spline.
- Create cubic spline log linear model of demand (NHPP model).
- F-test the cubic spline to determine significance.
- If p-value < .1 then do
  - Create Time homogeneous Poisson for use.
- End
- L=length of motif C_k
- For subsequences of length L
  - Create X^2 value.
  - Create associated p-value.
- End

- //Remove membership from motifs if significantly different from Poisson model.
- N=list of subsequences with membership in C_k
- For SS in N
  - If SS not significantly dissimilar from the Poisson process of C_k
    - Keep SS membership
  - Else
    - Remove SS membership from C_k
  - End
- End

End

//Add new members
Select least dissimilar motif for each subsequence
For each subsequence not significantly dissimilar from the closest motif add to LIST
Do while |LIST|>0
- Choose least dissimilar subsequence SS*
- Add SS* to associated motif C_*

Remove SS* and any subsequences which have overlap with SS* from LIST.
End

Phase 3-Generalized approach (Multivariate)

Remove motfs with membership size < MIN_SIZE_P3>
Let M be the set of motifs remaining
For C_k in M
Remove membership from motifs if significantly different from distribution.
N=list of subsequences with membership in C_k
For SS* in N
Calculate motif distribution for N-SS*
Determine F-like test statistic for SS*, using the N-SS* distribution.
Degrees of freedom both equal to length(C_k)*Number of factors-1,
If SS not significantly dissimilar from N-SS*
Keep SS membership to C_k
Else
Remove SS membership from C_k
End
End
If |N|>0, recreate motif centroid.
Of |N|=0, use motif centroid value from end of P2P2.
L=length of motif C_k
For subsequences of length L
Create F-like test statistic.
Create associated p-value.
End
End
//Add new members
Select least dissimilar motif for each subsequence
For each subsequence not significantly dissimilar from the closest motif add to LIST
Do while |LIST|>0
Choose least dissimilar subsequence SS*
Add SS* to associated motif C_*
Remove SS* and any subsequences which have overlap with SS* from LIST.
End

Phase 4

For motifs of size greater than 0

```
        Count number of motif occurrences for each time series
        Count length of each time series
        Relative occurrence=Occurrence Count/Length of series
End
//Cluster time series on feature space of relative motif occurrence
If <CLUS_PROC>=1
        Hard K-means produces results using PROC FASTCLUS
Else If <CLUS_PROC>=2
        Fuzzy C-means produces results
Else If <CLUS_PROC>=3
        Gath-Geva produces results using Fuzzy C-means as a centroid initializer.
End
```

APPENDIX B – SETTINGS FOR RUNS

Table B.1 - Parameter Settings for Application Runs

Number Variables	Time Series/ Application	B	Motif Window		Max Iterations		Motif Size Threshold	Goodness of Fit Threshold	Number of Clusters
			Min	Max	P1	P2			
1	Grocery/ Top Qty/Cross Cat	5	4	10	400	200	1	0.2	5
1	Grocery/ Top vs Mid Qty/ Beer	5	4	10	500	200	1	0.2	2
1	Billiards/ Strokes per Shot	3	7	16	500	200	5	0.1	5
1	Energy/ Naïve Approach	6	4	12	500	200	5	-	4
1	Energy/ Fractured Approach	6	20	24	500	200	5	0.02	2
2	Grocery/ Paired Sales Qty	6	4	10	500	200	8	0.05	2
2	Grocery/ Price and Promotion	6	4	10	500	200	5	0.1	3
2	Billiards/ Strokes and Chalk	1	7	20	500	200	8	0.2	4
3	Billiards/ Strokes,Chalk, Shot	1	7	20	500	200	2	0.3	4

Note that Hard K-Means was chosen as the preferred method for each of these examples, but that fuzzy c-means and Gath-Geva was considered for the billiards stroke/chalk/shot study and the grocery univariate top vs. mid quantity beer sales study.

APPENDIX C – TIME SERIES EXPLANATIONS FOR GROCERY APPLICATIONS

Table C.1 - Products/SKUs used in Univariate Grocery Study of Top Sellers

Category	Product Description	Product SKUs with same Sizing	Time Series Name
BEER	MILLER LITE BEER	3410057306	T11
BEER	OLD STYLE BEER	7336011301	T12
BEER	BECK'S REG BEER NR B	8248812345	T13
SOFT DRINKS	PEPSI COLA N/R	1200000230	T21
SOFT DRINKS	COCA-COLA CLASSIC	4900000639	T22
SOFT DRINKS	SEVEN-UP N/R	7800000034	T23
CHEESES	KR PHILA CREAM CHEES	2100061223	T31
CHEESES	KR AMERICAN SINGLES	2100060464	T32
CHEESES	DOM CREAM CHEESE	3828153081	T33
CEREALS	CHEERIOS	1600066610	T41
CEREALS	KELLOGGS CORN FLAKES	3800000120	T42
CEREALS	KELLOGGS FRUIT LOOPS	3800001720	T43
TOOTHPASTES	CREST TRT REG	3700000391	T51
TOOTHPASTES	CLGT REG	3500050900	T52
TOOTHPASTES	CREST TRT GEL	3700000309	T53

Table C.2 - Products/SKUs used in Univariate Grocery Study of Top vs. Mid Sellers

SELLING TIER	Product Description	Product SKUs with same Sizing	Time Series Name
TOP	MILLER LITE BEER	3410057306	T11
TOP	MILLER GEN DRFT LNNR	3410017505	T12
TOP	OLD STYLE BEER	7336011301	T13
TOP	BECK'S REG BEER NR B	8248812345	T14
TOP	HEINEKEN BEER N.R.BT	7289000011	T15
TOP	SAMUEL ADAMS LAGER N	8769210012	T16
TOP	MILLER SHARP'S N/A L	3410010505	T17
MID	PILSNER URQUELL NR B	7231163001	T21
MID	OREGON BREWERY NUTBR	79709634767	T22
MID	STROHS BEER N.R.BTLS	7204000006	T23
MID	BUDWEISER DRY BEER	1820000202	T24
MID	MICHAEL SHEA'S IRS A	7031033006	T25
MID	BECK'S OKTOBERFEST	8248812910	T26
MID	OLD STYLE ICE LN NR	7336011528	T27

Table C.3 - Products/SKUs used in Multivariate Grocery Study of Paired Sales

Category	Product Description	Product SKUs with same Sizing
BEER	BUDWEISER BEER	1820011168
		710
BEER	BUDWEISER LIGHT BEER	1820053168
		1820053178
		712
BEER	MILLER GENUINE DRAFT	3410017306
		3410015306
BEER	MILLER LITE BEER	3410057306
		732
BEER	OLD STYLE BEER	7336011301
BEER	OLD STYLE LT BONUS 6	7336097301
		7336097305

Table C.3 Continued

BEER	COORS BEER	7199011600
BEER	COORS LIGHT BEER	7199031600
BEER	BUSCH BEER	1820061168
		1820061166
BEER	BUSCH LIGHT BEER	1820086167
BEER	STROHS BEER	7204031310
BEER	STROHS LIGHT BEER	7204061321
SOFT DRINKS	PEPSI COLA CANS	420
		1200000017
SOFT DRINKS	PEPSI DIET CANS	421
		1200000053
SOFT DRINKS	COKE CLASSIC CANS	450
		4900001278
		4900000070
SOFT DRINKS	DIET COKE CANS	451
		4900001063
SOFT DRINKS	7-UP	440
		7800001214
SOFT DRINKS	DIET 7-UP	441
		7800001234
SOFT DRINKS	DR. PEPPER	442
		5490003019
SOFT DRINKS	DIET DR PEPPER	443
		5490073508
SOFT DRINKS	SPRITE	453
		4900000129
SOFT DRINKS	DIET SPRITE	454
		4900000459
SOFT DRINKS	MOUNTAIN DEW	426
		1200000088
SOFT DRINKS	DIET MOUNTAIN DEW	427
		1200000170

Table C.4 - Products/SKUs used in Multivariate Grocery Study of Pricing/Promotion

Category	Product Description	Product SKUs with same Sizing
BEER	BUDWEISER BEER	1820011168
		710
BEER	MILLER GENUINE DRAFT	3410017306
		3410015306
BEER	OLD STYLE BEER	7336011301
BEER	COORS BEER	7199011600
BEER	BUSCH BEER	1820061168
		1820061166
BEER	STROHS BEER	7204031310
SOFT DRINKS	PEPSI COLA CANS	*420*
		1200000017
SOFT DRINKS	COKE CLASSIC CANS	450
		4900001278
		4900000070
SOFT DRINKS	7-UP	440
		7800001214
SOFT DRINKS	DR. PEPPER	442
		5490003019
SOFT DRINKS	SPRITE	453
		4900000129
SOFT DRINKS	MOUNTAIN DEW	426
		1200000088

APPENDIX D – NULL RAND INDEX CALCULATIONS/EXPLANATION

D.1 Calculation of Null Rand Index:

Let the pair of time series $T_1, T_2, \ldots, T_N$ be classified into classes $C_1, \ldots, C_m$ which have volumes/sizes V_1 to V_m ($\sum_{k=1}^{m} V_k = N$). WLOG, let the first of the pair of compared time series be T_r and the second of the compared time series be T_s. Let RI_{NULL} be the null Rand index.

$$RI_{NULL} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP}{TP + TN + FP + FN} + \frac{TN}{TP + TN + FP + FN}$$

$$= \boldsymbol{P}(Guessing\ same\ class, being\ right) + \boldsymbol{P}(Guessing\ different\ class, being\ right)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} \boldsymbol{P}(Guessing\ same, being\ right | T_r \in C_i, T_s \in C_j)\, \boldsymbol{P}(T_r \in C_i, T_s \in C_j)$$

$$+\sum_{i=1}^{m}\sum_{j=1}^{m} \boldsymbol{P}(Guessing\ not\ same, being\ right | T_r \in C_i, T_s \in C_j)\, \boldsymbol{P}(T_r \in C_i, T_s \in C_j)$$

T_r and T_s are generic and no information about these is series specifically is known, the following simplification is possible:

$$\boldsymbol{P}\left(Guessing\ same, being\ right \middle| T_r \in C_i, T_s \in C_j\right) = \begin{cases} \boldsymbol{P}(Guessing\ same) if\ i = j \\ 0\ if\ i \neq j \end{cases}$$

$$\boldsymbol{P}\left(Guessing\ not\ same, being\ right \middle| T_r \in C_i, T_s \in C_j\right) = \begin{cases} \boldsymbol{P}(Guessing\ same) if\ i \neq j \\ 0\ if\ i = j \end{cases}$$

$\therefore RI_{NULL} =$

$$\sum_{i=1}^{m} \boldsymbol{P}(Guessing\ same)\, \boldsymbol{P}(T_r \in C_i, T_s \in C_i) +$$

$$\sum_{i=1}^{m}\sum_{j=1, j\neq i}^{m} \boldsymbol{P}(Guessing\ not\ same)\, \boldsymbol{P}(T_r \in C_i, T_s \in C_j)$$

$$=\sum_{i=1}^{m} \boldsymbol{P}(Guessing\ same)\, \boldsymbol{P}(T_s \in C_i \mid T_r \in C_i) \boldsymbol{P}(T_r \in C_i) +$$

$$\sum_{i=1}^{m} \boldsymbol{P}(Guessing\ not\ same)\, \boldsymbol{P}(T_s \sim \in C_i \mid T_r \in C_i) \boldsymbol{P}(T_r \in C_i)$$

$\boldsymbol{P}(Guessing\ same)$ and $\boldsymbol{P}(Guessing\ not\ same)$ are based on optimal strategy. Suppose we are given knowledge of number of clusters, m, which would be hypothesized, but not the associated volumes of those clusters. A reasonable assumption is that volume is assumed equal between clusters, i.e. V_{avg}=N/m. The total number of time series, N, is known. Without volumes to compute the actual values of $\boldsymbol{P}(T_s \in C_i \mid T_r \in C_i)\boldsymbol{P}(T_r \in C_i)$ and $\boldsymbol{P}(T_s\sim \in C_i \mid T_r \in C_i)\boldsymbol{P}(T_r \in C_i)$, these are computed by the guesser based on the theorized equal volumes of each class. RI_{HYP} is the hypothesized RI the guesser perceives as being obtained given a strategy on value of $\boldsymbol{P}(Guessing\ same)$. Let $\boldsymbol{P}(Guessing\ same)$ be denoted **P**(S). $\boldsymbol{P}(Guessing\ not\ same)$=1-**P**(S).

$$\therefore \mathrm{RI_{HYP}} = \sum_{i=1}^{m} \boldsymbol{P}(S)\frac{\mathrm{V}_{avg}-1}{N-1}\frac{\mathrm{V}_{avg}}{N} + \sum_{i=1}^{m}(1-\boldsymbol{P}(S)\left(\mathbf{1}-\frac{\mathrm{V}_{avg}-1}{N-1}\right)\frac{\mathrm{V}_{avg}}{N}$$

$$= \boldsymbol{P}(S)m\left(\frac{\mathrm{N/m}-1}{N-1}\right)\frac{\mathrm{N/m}}{N} + (1-\boldsymbol{P}(S)m\left(\frac{N-\mathrm{N/m}}{N-1}\right)\frac{N/m}{N}$$

$$= \boldsymbol{P}(S)\left(\frac{\mathrm{N/m}-1}{N-1}\right) + (1-\boldsymbol{P}(S)\left(\frac{N-\mathrm{N/m}}{N-1}\right)$$

$$= \left(\frac{1}{N-1}\right)\left(\left(\frac{2N}{m}-N-1\right)\boldsymbol{P}(S)+N-\frac{N}{m}\right)$$

Let $\mathbf{P}(\mathrm{S})^*$ be chosen so as to maximize $\mathrm{RI_{NULL}}$. Assuming N>1,

$$P(S)^* = \begin{cases} 1, \frac{2N}{N+1} \geq m \\ 0, \frac{2N}{N+1} < m \end{cases}$$

For N>1, and m≥2, which are all nontrivial case, $\mathrm{P(S)}^*$=0. This makes intuitive sense, because in the best case, there are two equal volume groups, and selecting the first time series out of the first group reduces the volume of that group, making choice from the second group

more likely. Thus, P(time series are different)>P(time series are same) and P(S)=0 maximizes the likelihood of guessing correctly.

Using this optimal strategy with the actual volumes and classes, the Rand index equation simplifies further.

$$\therefore \mathrm{RI}_{\mathrm{NULL}}(S^*) = \sum_{i=1}^{m} \boldsymbol{P}(T_s{\sim} \in C_i \mid T_r \in C_i)\boldsymbol{P}(T_r \in C_i)$$

This simplified result is used for the calculation of null model Rand indexes for the examples as given in Table D.1 below.

Table D.1 - Optimal Null Model Rand Indexes

Application	Section	Number of classes	Volume of Classes	Number of Time Series	Null Rand Index
Test Sets Univariate	2.6	8	10,15,5,1,1,1,1,1	35	0.731
Grocery/Top Sales	3.2	5	3,3,3,3,3	15	0.857
Grocery/Top vs Mid	3.2	2	7,7	14	0.538
Energy/Date	3.4	2	57,22	79	0.407
Energy/Day of Week	3.4	2	5,2	7	0.476
Test Set Multivariate	4.7	4	8,8,8,8	32	0.774
Grocer/Paired Sales	5.2	2	6,6	12	0.545
Grocery/Pricing Promotion	5.2	2	6,6	12	0.545

APPENDIX E – BILLIARDS STUDY INFORMATION

Table E.1 - Player Information

Player ID	Name	Handicap
P1	Eric Olson	7
P2	Chas Barlow	4
P3	Graham Latter	5
P4	Tommy Gooch	5
P5	Steve Swaim	7
P6	Rick Allen	5
P7	Peter Abatangelo	5
P8	Aqeelah Jones(AJ)	3
P9	Gabe Josset	5
P10	Dave Hansen	6
P11	Will Bass	7
P12	Amber Fraizer	5
P13	Kayla Peck	4
P14	Jenny Stoner	4
P15	Omar Budwan	4
P16	Matt Coats	5

Table E.2 - APA Equalizer® Handicapping System-Games to Win a Match By Handicap

		P2					
		2	**3**	**4**	**5**	**6**	**7**
P1	**2**	2/2	2/3	2/4	2/5	2/6	2/7
	3	3/2	2/2	2/3	2/4	2/5	2/6
	4	4/2	3/2	3/3	3/4	3/5	2/5
	5	5/2	4/2	4/3	4/4	4/5	3/5
	6	6/2	5/2	5/3	5/4	5/5	4/5
	7	7/2	6/2	5/2	5/3	5/4	5/5

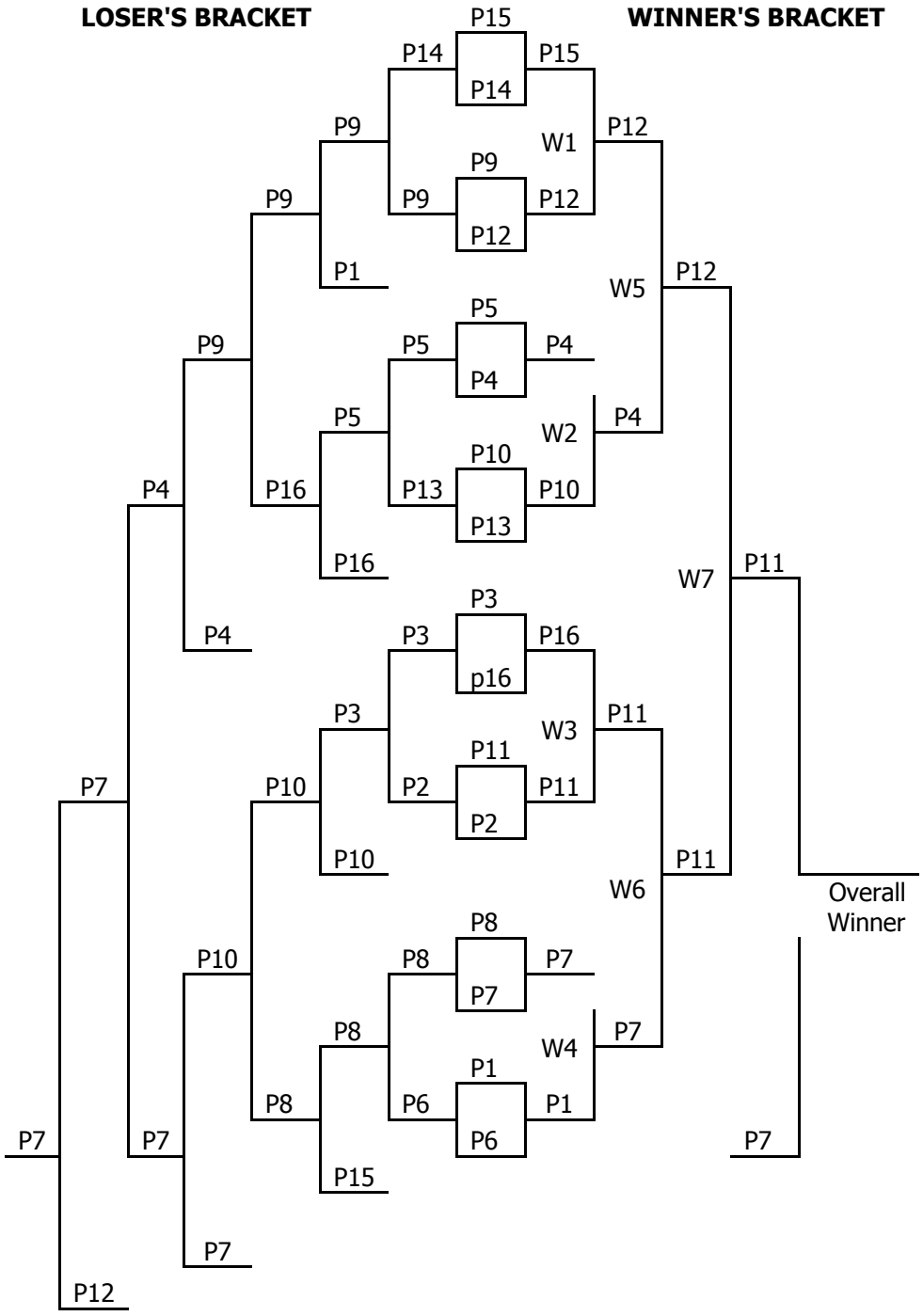


Figure E.1 - Results of Tournament (Double Elimination Style) Movement to the right in center indicates a player won, then movement away from center indicates a win.