Time Series Clustering
Using Copula-Based Higher Order Markov Process

by

Yuan Zhuang

(Under the direction of Nicole Lazar)

Abstract

In model-based time series clustering, most models used only allow linear dependence. This could lead to unsatisfactory clustering results due to the limited use of information. We propose using the copula-based higher order Markov process (CHOMP) by Ibragimov (2009). The CHOMP can capture not only dependence strengths, but also different dependence structures (linear or non-linear). We further relax the stationarity condition in the original version of CHOMP by Ibragimov (2009) so that it can also capture the profile/shape information in non-stationary time series. Moreover, a non-parametric estimation for the CHOMP is proposed based on the two-step procedure of Chen and Fan (2006). Finally, a time series clustering algorithm based on CHOMP is proposed using agglomerative hierarchical clustering and finite mixture models. With more information extracted and used in clustering, our algorithm outperforms its competitors in an extensive simulation study and two real data analyses.

Index words: cluster analysis, model-based, non-parametric, non-linear, dependence structure

Time Series Clustering
Using Copula-Based Higher Order Markov Process

by

Yuan Zhuang

B.S., Southwest Jiaotong University, Chengdu, China, 2010

M.S., University of Georgia, Athens, GA, 2012

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2015

Time Series Clustering
Using Copula-Based Higher Order Markov Process

by

Yuan Zhuang

Approved:

Major Professor:	Nicole A. Lazar
Committee:	Jeongyoun Ahn
	Lynne Billard
	Cheolwoo Park
	Jaxk Reeves

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2015

Time Series Clustering Using Copula-Based Higher Order Markov Process

Yuan Zhuang

October 25, 2015

Dedication

This dissertation is dedicated to my beloved wife and my dear parents.

Contents

List of Figures

List of Tables

Chapter 1

Acknowledgments

First and foremost, I would like to express my deepest appreciation to my dedicated advisor, Dr. Nicole A. Lazar. She has been such a great mentor by giving me inspiring guidance, indispensable suggestions, without which this dissertation would not have been possible. I am grateful for your constant encouragement and support accompanied me in all the time of research for and writing of this dissertation.

I want to take this opportunity to express my sincere gratitude to my committee members, Dr. Jeongyoun Ahn, Dr. Lynne Billard, Dr. Cheolwoo Park and Dr. Jaxk Reeves, for their patience, time, and suggestions in completing the dissertation. Not only have they provided guidance whenever needed, they have provided intellect.

I would like to extend my sincere thanks to all fMRI group members, for their valuable help and substantial consistent support throughout my study. I would like to thank Dr. Cheolwoo Park for providing me the resting state data. I want to give my appreciation to Dr. Jennifer McDowell and her students, Amanda, David, Jordan for all the suggestions and data they provided. I also want to thank Dr. Gery Geenens from University of New South Wales and

his collaborators for their codes. Many thanks also go to my fellow colleagues and friends here for their constant support and for providing an atmosphere conducive to research. I will truly enjoy to work with these fine people again.

Finally, I want to thank my parents and my wife Beatrice Zhang, I could not have accomplished this dissertation without their love and support.

Chapter 2

Introduction

The history of humans clustering things has almost the same length as the human history itself. Once we were able to cognitively notice the differences among things, we started the process of clustering. But it was not until recently that we systematically study the mathematical cluster analysis. According to Bailey (1994), modern cluster analysis originated in anthropology in the 1930's, and then in the late 1930's and early 1940's it was introduced to psychology where it started to receive more attentions. Nowadays cluster analysis is needed in almost every field of scientific endeavor, e.g. finance, biology, genetics and engineering. The clustering problems in these areas are also becoming more and more difficult. Fortunately, statisticians and scientists have developed a lot of powerful methods for clustering. But before going into those complex problems and sophisticated clustering methods, I would like to start with an easy clustering task.

Suppose one is given a bunch of apples and oranges, and is asked to cluster them. Well, this is pretty easy and most people will end up with two groups, one for apples and the other for oranges. But let us think twice: why is it so easy? What makes it easy?

Let us put it in another way. Suppose one is given the same apples and oranges, but we now put them behind a curtain so that one can only see their rough shapes. Then the clustering becomes more difficult because all other information are unavailable except for their shapes which are very similar. Therefore with only the shape information, it is difficult to cluster these objects. However, if one is allowed to touch them, then the clustering becomes easy due to the additional texture information. Further, if one can see them, it becomes even easier and one can cluster them almost immediately.

The idea this example conveys is that when one can obtain more useful information from the objects, clustering them becomes easier. Like clustering the apples and oranges, with more and more information available, it becomes easier and easer to distinguish apples from the oranges. Of course, after we obtain the useful information from the objects, we still need to employ appropriate algorithms to actually perform the clustering procedure. However, without the right information, more advanced algorithms won't help. Like in the apple and orange example, it is hard to cluster them simply based on the shape information. Even the human brain, the most complicated "computer" in the world, doesn't work very well. This gives the motivation of our work: to achieve better clustering by extracting more useful information from the data.

Time series clustering has raised a lot of interests in recent years, and has very important applications in various fields, such as signal processing, engineering, economics, medicine, art, entertainment and so on. See more examples of the application of time series clustering in Liao (2005). Based on the way information is extracted from the data, Liao (2005) claims that there are mainly three types of approaches for time series clustering: (1) clustering based on raw series; (2) clustering based on extracted features; (3) clustering based on statistical models. The first one is easy to understand because it is just regular clustering algorithms applied to the raw series, treating them as long vectors. The second approach

extracts information from the time series through one or more important features, such as the (partial) auto-correlation, linear slope and so on. The third approach represents the time series by statistical models and then clusters either based on model outputs (such as model coefficients and residuals) or through the finite mixture model approach (see Section 3.1 for more details). The latter two approaches are closely related because they are both trying to efficiently extract important information from the data. They just do that through different ways. In this work we focus on the third approach where the information in the time series data is represented and extracted by statistical models.

As seen in the apple and orange example, having more useful information available could make clustering easier. Then what kind of information can be collected from the time series, as analogy to the textures and colors for the fruits? First, one can have shape or profile information of the series. If we treat a time series as a sequence of random variables, the series profile is actually determined by a sequence of marginal distributions. With these marginals, one can not only account for the sequence profile through the marginal means, but also the variabilities at each point through the marginal standard deviations (s.d.). The second type of information we can have is the intrinsic temporal dependence of the series. This might be a little hard to imagine, but usually a time series x_t can be decomposed as $x_t = \mu_t + \epsilon_t$, where μ_t carries the profile information and ϵ_t carries the intrinsic temporal dependence information. In other words, the intrinsic temporal dependence of a series, x_t, is the dependence information of the same series with its profile information removed.

Moreover, within the temporal dependence there is the dependence strength and the dependence structure. Different structures of temporal dependence can lead to very different time series behaviors even if their dependence strengths are the same (see Section 5.2.3 for examples).

Ideally, we would want to collect complete information from the time series, and use all of them for clustering. However, in the literature, information about temporal dependence is usually not completely used in clustering. The key reason for that lies in the model.

In model-based time series clustering, researchers have used ARIMA models, dynamic regression models, GARCH models, and Markov models. More details about these models and a literature review will be given in Section 3.2. However, none of these model collects complete information from time series. For example, the dynamic regression model, ARIMA model, and GARCH model assume linear dependence and normality in the time series. Therefore these models only capture the dependence strengths inside the linear and Gaussian framework. The strength of non-linear dependence and the dependence structures cannot be accurately captured. Markov models, on the other hand, allow for non-linear and non-Gaussian dependence, but only apply to discrete time series with first-order dependence (current observation depends only on the previous one). Therefore we need a model that will provide us more complete information from the time series, hence benefit the clustering.

We propose using the copula-based higher order Markov process (CHOMP). This model is a discrete-time Markov process originally proposed by Ibragimov (2009). It has several advantages over other models used in the literature: (1) it can capture the strength of both linear and non-linear dependence. (2) It can capture the dependence structure. (3) It is able to model real-valued time series as it has continuous state space. The original version of CHOMP proposed by Ibragimov requires the data to be stationary. In other words, the original CHOMP cannot capture the profile information in time series if any. In this work, we relax the stationarity assumption in a modified version of CHOMP such that it can handle non-stationary time series with arbitrary profiles. This contribution is presented in Corollary 5.7 of Section 5.2. Moreover, we propose a non-parametric estimation of the CHOMP based on the two-step procedure by Chen and Fan (2006) in Section 5.3. With this contribution,

the CHOMP model will not have any distributional assumption and can handle a larger variety of dependence structures.

Based on the modified non-parametric CHOMP, we propose a model-based time series clustering algorithm using agglomerative hierarchical clustering and finite mixture model. With CHOMP's ability to extract more information, the proposed algorithm has the potential to provide a better clustering of time series. In fact, in an extensive simulation study and two real data analyses, we find that, our algorithm outperforms its competitors in most situations, and performs equally well in other situations. Moreover, even the number of clusters becomes easier to choose using our algorithm in the second real data analysis where we cluster resting state fMRI time series. These results show further support to our argument in the apple and orange example: using more useful information makes clustering better and easier. Therefore we believe that our model-based clustering algorithm using modified CHOMP is a very attractive alternative in clustering time series.

The remainder of this work is organized as follows. Chapter 3 gives a brief literature review on general clustering algorithms and model-based time series clustering. Since we are using the copula tool in the CHOMP model, Chapter 4 briefly introduces the copula theory and discusses the dependence structure exhibited by different copula functions. Then in Chapter 5, we discuss the CHOMP model in more details and show how we obtain the modified CHOMP by relaxing the stationarity assumption in the original CHOMP by Ibragimov (2009). Chapter 5 also explains how the dependence structure in copula functions is reflected in the time series generated by CHOMP. In Chapter 6, we give the actual clustering algorithm using modified CHOMP for both stationary and non-stationary time series. Chapter 7 and Chapter 8 present an extensive simulation study and two real data analyses, respectively. Finally, a discussion is given in Chapter 9.

Chapter 3

Literature Review

In this Chapter, we will review the literature on model-based time series clustering. Specifically, in Section 3.1, we give an overview of the general clustering algorithms for traditional data points and vectors. In Section 3.2, we review the literature on model-based time series clustering. Finally, in Section 3.3, we explain the general ideas and advantages of our proposed clustering algorithm using CHOMP.

3.1 General Clustering Algorithms

Most popular clustering algorithms are based on three classes of methods: hierarchy-based clustering, iterative relocation partitioning and model-based clustering.

Modern hierarchy-based clustering algorithms originate from agglomerative hierarchical clustering (Florek et al. 1951a, 1951b as cited in Dawson and Belkhir 2009). This algorithm starts with every object as a cluster and successively combines objects or groups of objects based on their distance information (closest objects or groups of objects are combined to

form new groups). More specifically, for a data set with n data points, agglomerative hierarchical clustering begins with n clusters (each point is its own cluster). After calculating all the pairwise distances, the two closest points are combined to form a new cluster. The total number of clusters then reduces to $n - 1$. This combining step repeats until all the points are combined. In other words, the clustering algorithm ends when only one cluster is left. However, a clustering result with one cluster is usually not desired. Thus it is crucial to determine where the combining should stop, or equivalently, what is the right number of clusters. The most common way to determine the number of cluster k is through calculating certain clustering evaluation criteria for each possible value of k. Then the most appropriate k is achieved by optimizing the value of the selected criterion. Commonly used criteria include the silhouette width, the Dunn index and so on. See Section 6.1 and Brock et al. (2008) for more details about these criteria. Variants of agglomerative hierarchical clustering include divisive hierarchical clustering (MacNaughton-Smith et al. 1964) and pyramidal symbolic clustering (Diday et al. 1993).

The second class of algorithms is based on the iterative relocation partitioning. As the name indicates, this class of algorithms partitions the data set by iterative relocation. In these algorithms the membership of each object, and sometimes each cluster center, is updated iteratively based on some optimality criterion. The most famous representative in this class is k-means clustering (MacQueen 1967). K-means clustering relocates each object to the group whose center is the closest, and this is done repeatedly until no further relocation can be made. More specifically, if given a data sets with n data points, k-means clustering first needs to know the number of clusters k. Then it randomly partitions the data points to k clusters, and the center of each cluster is usually calculated as the average of all the points in that cluster. Then the distances between each point and each center are used for relocation. The relocation step in k-means clustering is to move each data point to the cluster that

is closest to it. The closest cluster is determined by comparing the distances to all cluster centers. After all data points have been relocated, the cluster centers are also updated based on the newly formed clusters after relocation. Then based on the new cluster centers, the distances to each data points can be calculated again and the next iteration of relocation proceeds. The algorithm ends when no new relocations can be made. Unlike agglomerative hierarchical clustering, k-means clustering needs the number of clusters, k, and an initial clustering assignment in order for the algorithm to start running. The number of clusters can also be determined through optimizing some clustering criterion over all possible values of k. Variants of k-means clustering have also been developed to extend the technique to different situations. For example, fuzzy c-means (Dunn 1973) uses fuzzy distances, so each data point doesn't fully belong to a cluster. The k-medoids algorithm (Kaufman and Rousseeuw 1987) deals with situations in which the average may not be the best center of a cluster.

The third class of algorithm is the model-based clustering which often refers to the finite mixture model approach. This type of algorithm is frequently used in model-based time series clustering. Some of the early research on using the finite mixture model in clustering was proposed in late 1980's and early 1990's (McLachlan and Basford 1988; Banfield and Raftery 1993; Cheeseman and Stutz 1995). Fraley and Raftery (2002) give a good review on using finite mixture model for clustering. Its general idea assumes a probability model for each cluster, so the entire data set is assumed to be from a mixture of several probability models. Then the clustering problems can be regarded as a statistical model selection problem, since different clustering results stand for finite mixture models with different components and/or different number of components. Specifically, let us assume the n data objects, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$, are generated by M different probability models that correspond to the M clusters of interest. Let $z_{ik} = \{1, 0\}$ be the cluster indicator for each data object and each cluster, e.g. $z_{1,3} = 1$

means that object 1 belongs to cluster 3 (the third probability model). Let $l(\mathbf{x}_i|\mathbf{\Phi}_k)$ be the conditional likelihood that data object $\mathbf{x}_i$ is generated from the k-th probability model with parameter set $\mathbf{\Phi}_k$. Let p_k be the prior probability that a data object is from the k-th model such that, each $\mathbf{z}_i$ is i.i.d. from a multinomial distribution of one draw from M categories with probabilities, $p_1, ..., p_M$. The conditional log-likelihood of the entire mixture model can then be expressed as

$$L((\mathbf{x}_1, ..., \mathbf{x}_n)|\Theta) = \sum_{i=1}^{n} \sum_{k=1}^{M} z_{ik} \log[l(\mathbf{x}_i|\mathbf{\Phi}_k)p_k]$$

where $\Theta = \{\Phi_1, ..., \Phi_M, p_1, ..., p_M, z_{1,1}, ..., z_{n,M}\}$ is the parameters set for the mixture model, and $\sum_{k=1}^{M} p_k = 1$. One needs to find $\hat{\Theta}$ such that

$$\hat{\Theta} = arg \max_{\Theta} L((\mathbf{x}_1, ..., \mathbf{x}_n)|\Theta)P(\Theta)$$

When one uses a non-informative prior on Θ, then $\hat{\Theta}$ becomes maximum likelihood estimation. The usual expectation-maximization (EM) algorithm is often employed to get the estimator treating $\{z_{1,1}, ..., z_{n,M}\}$ as the unobserved data.

To choose the number of clusters in the finite mixture model setting, different types of model selection methods in the clustering context have been proposed. This is because choosing the number of clusters is the same as selecting between two mixture models with different number of components. Since we are able to calculate the likelihood values for the mixture models, many statistical techniques are available to choose the number of clusters. McShane et al. (2002) and Liu et al. (2008) use the finite parametric mixture model. Once they have a clustering result, they use bootstrap to repeatedly test if one of the clusters should be further divided into two. They achieve the optimal number of clusters if none of the derived clusters should be further divided. Maitra and Melnykov (2010a) have developed a *quantitation map*

in parametric finite mixture models. It can help researchers compare a simpler mixture model with a more complicated one through a hypothesis test. A simpler mixture model here means a smaller number of clusters. Their work has been further extended by Maitra et al. (2012) to nonparametric settings. Bayesian infinite mixture model has also been proposed. It is able to automatically select the number of clusters by using a Dirichlet process as the prior for the parameters in the finite mixture model. Then the optimal number of clusters can be chosen through the Monte Carlo Markov Chain (MCMC) procedure. Interested readers about this infinite mixture model are referred to Ferguson (1973) and Hjort et al. (2010) and the references therein for more complete explanations.

Both hierarchy-based clustering and iterative relocation partitioning are easy to implement in practice, which may be the most important reason why they are so popular. But they both have drawbacks as well. For example, in agglomerative hierarchical clustering, if a data point is connected with another one, they can never be separated even if they are later found to be from different clusters. This inflexibility sometimes makes it harder for the algorithm to reach the optimal clustering. See Herrero et al. (2001) for more about the drawbacks of hierarchical clustering. The iterative relocation algorithms, on the other hand, have the problem of selection of the initial clustering assignment as mentioned in Pena et al. (1999). The final clustering result depends strongly on the initial clustering assignment. Most of the time, only when the initial assignment is relatively close to the true one, can the algorithms achieve the best clustering assignment. Otherwise, the result is only locally optimal, meaning it is only the best among those assignments close to the initial one. Moreover, as mentioned by Fraley and Raftery (2002), most algorithms in these two classes are heuristically motivated. That means, they are easy to understand and sometimes easy to implement, but their statistical properties are largely unknown.

The model-based clustering algorithm doesn't have these problems. It has a complete statistical characterization which makes it easier for researchers to study its statistical properties. As mentioned earlier, many statistical approaches have been proposed to select the number of clusters. Nevertheless, the model-based clustering algorithms are usually slower in computation, and harder to understand and implement. Also, as we mentioned in the apple and orange example in Chapter 2, if we don't collect more useful information, the clusters could appear similar to each other at the starting point. For example, with only the shape information, apples and oranges appear similar. When that happens, using more sophisticated clustering algorithms won't help because it is very hard to cluster based on little difference among the groups.

3.2 Model-Based Time Series Clustering

Recently, more interest has been raised in applying cluster analysis to more complicated data, such as time series data, interval data, distribution-valued data and so on. These data are sometimes called symbolic data or object-oriented data. Our interest is on time series clustering. According to Liao (2005), there are three main types of approaches in time series clustering: (1) clustering based on the raw time series; (2) clustering based on the extracted features; (3) clustering based on statistical models. In this work, we are interested in clustering time series based on statistical models for the following reasons. First of all, representing time series with statistical models can be efficient in summarizing redundant data. For example, when the time series is long but has consistent pattern, fitting a model reduces the complexity of the data by keeping the pattern information only, whereas the raw series approach has to keep the whole series. Secondly, a lot of time series have a high level of noises, in which case it is better to use distribution-based statistical models to

characterize the series. Appropriate statistical models allow one to represent the time series in a way that information from both the signals and the noise can be nicely combined and extracted. Nevertheless, this is usually difficult for the other two approaches. Using the raw series approach, the signal in the series can easily be overwhelmed by the huge amount of noise. Using the feature extraction approach, one can only focus on a limited number of features and choosing which features to use is still a problem. Also, for describing the noise, distribution-based characterization is usually more appropriate than certain deterministic measurements used in feature extraction. A more comprehensive comparison among these three approaches is also needed, but is not within our discussion.

In this Section, we will briefly review the literature of model-based time series clustering. Researchers have developed quite a few approaches in this field by combining the general clustering algorithms with statistical models. Two primary types of combining approaches used are: distance-based algorithm and finite mixture model.

The distance-based algorithm fits a statistical model to each time series separately and then applies traditional cluster analysis to the model outputs, e.g., model parameters, residuals, transition probability matrices and so on. For example, Piccolo (1990) assumes an ARIMA model for each time series and uses the Euclidean distance between two sets of model parameters as the distance measure in agglomerative hierarchical clustering. Maharaj (2000) assumes a stationary AR(p) model for time series, and repeatedly tests whether the AR parameters from two series are equal or not. Then the p-values from those tests are used as the pairwise distances in agglomerative hierarchical clustering. Baragona (2001) uses the ARMA model assumption for each time series, but he calculates the cross-correlation of the model residuals as the distance between two series. He then employs several searching algorithms, including Tabu search and the genetic algorithm, to reach the optimal partition based on some criterion specifically designed for his data. Kalpakis et al. (2001)

assume ARIMA model for the data. Each time series is represented by the linear predictive coding(LPC) cepstrum (inverse fourier transformation of the logarithm of the estimated spectrum of a signal), discrete fourier transformation (DFT), discrete wavelet transformation (DWT), principal component analysis (PCA), and the auto-correlation function. Then the k-medoids algorithm is performed using the Euclidean distance. Kalpakis et al. (2001) find in his simulation results that the LPC cepstrum provides higher discriminatory power and better clustering results than all other methods for ARIMA time series. This method can also be considered a feature extraction approach. We list it here because it also assumes an ARIMA model for the time series.

The second type of model-based approach for clustering time series is based on the finite mixture model. The general idea is very similar to what is discussed in Section 3.1 except for that now we use time series models instead of general probabilistic models. The models that have been employed in this type of approach include ARIMA models, Markov models, dynamic regression models and GARCH models.

Xiong and Yeung (2002) propose a finite ARIMA mixture model for clustering time series data. The data are assumed to be generated by a finite number of ARIMA models with unknown membership. An expectation-maximization (EM) algorithm is derived to learn the mixture probabilities as well as the parameters of each ARIMA model. By running the EM algorithms multiple times, the number of clusters is determined by identifying redundant components in the mixture model. However, their approach can only be applied to ARIMA time series which only allows for linear dependence structure.

The Markov chain mixture model has also been studied (Poulsen 1990; Ridgeway 1997; Smyth 1999; Cadez et al. 2000a, 2000b, 2000c), and the EM algorithm is commonly used by researchers to find the optimal clustering in this context. In addition, Ramoni (2002) proposes Bayesian clustering by dynamics which is a combination of agglomerative hierar-

chical clustering and Markov chain. It finds the optimal clustering result by merging the two closest Markov chains repeatedly. The Kullback-Leibler divergence (Kullback and Leibler 1951) of the two probability transition matrices serves as the distance measure. The selection of the number of clusters is treated as a Bayesian model selection problem and determined by maximizing the posterior probability. In other words, if the clustering after merging two Markov chains is less probable than the one before, the algorithm stops and the number of clusters is automatically achieved. Using a Markov chain allows them to account for more complicated dependence structure in time series than ARIMA, but it only considers the first order dependence. That is, given the value of the previous random variable in the chain, the current random variable is independent of all others prior to that. Also, most work only considers discrete-valued Markov chain.

The finite mixture model based on hidden markov model(HMM) has also been employed by some researchers (Krogh et al. 1994; Owsley et al. 1997) with EM algorithm applied to find the optimal clustering. Alternatively, k-means clustering (Perrone and Connell 2000) and the rival penalized competitive learning (RPCL) algorithm (Law and Kwok 2000) have also been used in place of the EM algorithm. The number of clusters can be determined by Monte-Carlo cross-validation (Smyth 1997) and information criteria like the Bayesian information criterion (BIC) (Li and Biswas 2000). But the problem is that, same as the Markov chain, most of these works using HMM apply only to the discrete state space. Li and Biswas (1999) propose a HMM-based algorithm which claims to be able to handle continuous data. However, the core model in the algorithm still assumes a discrete state space. Continuous values need to be converted to discrete values through a pre-processing step to enter the core model.

More recently, Frühwirth-Schnatter and Kaufman (2008), Frühwirth-Schnatter (2011) and Juárez and Steel (2010) use the dynamic regression model which is a generalized form of

AR(p). It has the same autoregressive terms as AR(p), but has additional regression terms to account for other effects. Frühwirth-Schnatter (2011) provides a review on using this model for time series clustering. The clustering algorithm these authors use is the Bayesian finite mixture model which estimates the model parameters through Markov Chain Monte Carlo (MCMC) instead of the EM algorithm. Luan and Li (2003) and Heard et al. (2006) proposes using non-linear functions to model the profile of genetic sequence data and adding random effects which allow sequences to vary within the same cluster. They use the regular finite mixture model setting and estimate the parameters by the EM algorithm. Bauwens and Rombouts (2007) also consider using GARCH models and they also use the Bayesian finite mixture model mentioned by Frühwirth-Schnatter (2011). Nieto-Barajas and Contreras-Cristán (2014) use a special form of the dynamic regression model as their model for time series. But instead of the regular finite mixture model, they use a Poisson-Dirichlet process mixture model to perform clustering. This mixture model is a more general case of the Dirichlet process mixture model mentioned in Section 3.1, and has the same advantage of selecting the number of clusters. What is interesting is that, according to Frühwirth-Schnatter (2011), all of the models mentioned in this paragraph, can be treated as a special case of the dynamic regression model.

We summarize this rich literature by asking two important questions in clustering. First, what time series model is used and what information is captured for clustering? Second, what algorithm is used to cluster this information?

The time series models used in the literature include: (1) ARIMA models; (2) dynamic regression models; (3) Markov models. However, none of them is able to extract complete information from the time series data. Dynamic regression model and ARIMA models can be used for real-valued time series, but they can only capture the strength of linear dependence due to the linear form in the models. The non-linear approach used by Luan and Li (2003)

and Heard et al. (2006) is only non-linearly extracting the profile information of the time series, not the intrinsic temporal dependence. The Markov model is able to capture the strength and structure of dependence, but it only applies to discrete-valued time series and can only extract first-order dependence. Therefore a clustering algorithm using a model that can capture more complete temporal dependence information in real-valued time series is lacking. We propose a clustering algorithm using the copula-based higher order Markov process (CHOMP) and I believe our algorithm is able to fill the gap. We give the general idea of our proposed algorithm in Section 3.3 and details in the rest of the dissertation.

For the second question, the clustering algorithms used in the literature include: (1) the simplest distance-based algorithms using heuristic clustering methods (like k-means clustering and agglomerative hierarchical clustering); (2) the finite mixture model and its Bayesian counterpart; (3) Bayesian infinite mixture model (Dirichlet process mixture model). In our proposed algorithm, we use both the first and the second approach.

3.3 Copula-Based Higher Order Markov Process

As discussed in the last Section, the current models used in model-based time series clustering have limitations. They cannot collect complete information from the time series data, which could potentially lead to failure or inefficiency in clustering. Just like in the apple and orange example, clustering suffers when only given the shapes of the fruits. To make it more clear, we summarize previous models' drawbacks as follows: (1) they cannot capture the strength of non-linear dependence. For example, both the ARIMA models and the dynamic regression models only allow linear additive temporal dependence. Since these models assume the dependence all have the same structure (linear structure), they are, in fact, only capturing the dependence strength. (2) they cannot capture the dependence structure. (3) Models like

Markov chain and HMM don't accept real-valued time series. (3) Many statistical models used are parametric. For example, dynamic regression models and ARIMA models have the normality assumption. Therefore methods using non-parametric or semi-parametric models are needed.

To address the mentioned problems, we propose a time series clustering algorithm using the copula-based higher order Markov processes (CHOMP). This model is a discrete-time Markov process originally proposed by Ibragimov (2009). It is able to capture more complete temporal dependence information in time series. It is also able to model real-valued time series as it has a continuous state space. The original version of CHOMP proposed by Ibragimov (2009) requires the data to be stationary. We relax the stationarity assumption such that we have a modified CHOMP that can also handle certain non-stationary time series. This contribution is presented in Corollary 5.7 of Section 5.2. As a result, the modified CHOMP can be fully determined by a copula function and a group of univariate marginal distributions. The copula function characterizes the temporal dependence, and the marginal distributions characterize the profile information. For the marginal distributions, we use an adjusted empirical cumulative distribution function (c.d.f.) so that the modified CHOMP can model non-stationary time series with arbitrary profile. See Section 5.3 for more details about the marginal estimations. For the copula function, we use both parametric and non-parametric copulas. When using parametric copulas, the modified CHOMP becomes a semi-parametric model that can capture dependence strength of certain non-linear dependence. When using non-parametric copulas, the modified CHOMP becomes a non-parametric model that can capture not only the dependence strength (linear or non-linear), but also the dependence structure. See Section 4 and Section 5.3 for more details about the copulas and the estimation of the modified CHOMP. Therefore the modified CHOMP are able to capture the more complete temporal dependence information than models previously used from time series.

Our proposed algorithm combines the modified CHOMP with both the distance-based approach and the finite mixture model approach mentioned in Section 3.2. For the distance-based approach, we fit a CHOMP to every time series and the estimated copula functions or copula parameters are used as the model outputs for clustering. We then apply agglomerative hierarchical clustering to cluster these model outputs. The number of clusters is determined by optimizing certain internal validity measures. This algorithm is suitable for clustering stationary time series. We will give more details about it in Section 6.1. To cluster non-stationary time series, we use the finite mixture model approach. All time series are assumed to be generated by a finite mixture of modified CHOMP models. To estimate such a mixture model, we use an adaptation of the classification expectation maximization (CEM) proposed by Celeux and Govaert (1992). The CEM algorithm combines ideas from both the EM algorithm and iterative relocation. It still uses the setting of finite mixture model, but adds a classification step between the E-step and the M-step of the regular EM algorithm. Specifically, suppose we have n time series. We start with an initial assignment of k clusters. In each cluster, a CHOMP is fitted and the log-likelihoods of each time series to each of the k estimated CHOMPs are obtained. This is the new E-step. The log-likelihood values are then used in the classification step where each time series is relocated to the cluster in which the fitted CHOMP gives the highest log-likelihood. Then with the new cluster assignment after relocation, the CHOMP is fitted again in each of the new clusters. This is the new maximization step. Then based on the newly estimated CHOMPs and clustering, the E-step starts again. The iteration continues until no new relocation can be made. The optimal number of clusters can be achieved through a Bayesian model selection procedure which minimizes the Bayesian information criterion (BIC). More details about the second algorithm are given in Section 6.2.

The most important advantage of our proposed clustering algorithm over other algorithms in the literature is that it is able to capture more useful information from the time series, including the dependence strength, dependence structure and sequence profile. This augmented information will improve the efficiency and accuracy of the clustering. Just like the apple and orange example, with more information (colors and textures) available, the clustering becomes better and easier. This argument is further supported by our results in an extensive simulation study and two real data analyses in Chapter 7 and Chapter 8, respectively.

Chapter 4

Copula Theory

As mentioned in Chapter 2 and Chapter 3, the advantages of the CHOMP over other models used in time series clustering are: (1) it is able to extract more information from the time series. It is not only able to capture the strength of linear and non-linear dependence, but also the dependence structure. (2) It applies to real-valued time series. (3) It can be a semi-parametric or even a non-parametric model which does not have any distributional assumption. In fact, these advantages of the CHOMP are largely due to its use of the copula. Therefore we think it is necessary to at least give a brief introduction to the basics of copula theory. Interested readers are encouraged to read the paper by Embrechts et al. (2003) and the book by Nelsen (2006) for more details.

The organization of this Chapter is as follows: we first give the definitions of copula and tail dependence in Section 4.1. Then we introduce the major parametric copula families in Section 4.2, and several non-parametric copulas in Section 4.3.

4.1 Definition

A formal definition of a copula is given in Nelsen (2006):

Definition 4.1. *A function* $C : [0,1]^n \to [0,1]$ *is called an* n*-dimensional copula if it satisfies the following conditions:*

- $C(u_1, ..., u_n)$ *is an increasing function in each component* u_i*.*
- $C(u_1, ..., u_{k-1}, 0, u_{k+1}, ..., u_n) = 0$ *for all* $u_i \in [0,1], i \neq k$*, and all* $1 \leq k \leq n$*.*
- $C(1, ..., 1, u_i, 1, ..., 1) = u_i$ *for all* $u_i \in [0,1]$*, and all* $1 \leq i \leq n$*.*
- *For all* $(a_1, ..., a_n), (b_1, ..., b_n) \in [0,1]^n$ *with* $a_i \leq b_i$*,*

$$\sum_{i_1=1}^{2} ... \sum_{i_n=1}^{2} (-1)^{i_1+\cdots i_n} C(x_{1i_1}, ..., x_{ni_n}) \geq 0$$

where $x_{k1} = a_k$ *and* $x_{k2} = b_k$ *for all* $1 \leq k \leq n$*.*

The last condition in Definition 4.1 is sometimes called n-increasing and generalizes the one-dimensional non-decreasing property to higher dimensions. It is basically saying that a copula is non-negative in every n-dimensional sub-hypercube of $[0,1]^n$. In addition, an n-dimensional copula itself can be regarded as a joint c.d.f of n random variables, $U_1, ..., U_n$, each of which is uniformly distributed on $[0,1]$.

Statisticians then utilize this special function to connect between the joint distribution and the marginals. Specifically, given any set of random variables $\{X_1, X_2, ..., X_n\}$, a copula or copula function, C, can be used as a multivariate link function between their marginal cumulative distribution functions (c.d.f.), $F_1(x_1), ..., F_n(x_n)$, and the corresponding joint c.d.f., $F(x_1, ..., x_n)$. That is,

$$F(x_1, ..., x_n) = C(F_1(x_1), ..., F_n(x_n)) \tag{4.1.1}$$

where each x_i is the realization of the corresponding random variable, X_i. In copula theory, $F_i(x_i)$ is usually denoted by u_i. So Equation 4.1.1 is usually written as

$$F(x_1, ..., x_n) = C(u_1, ..., u_n)$$

We can do so due to the special properties defined in Definition 4.1.

Sklar's theorem (Sklar 1959) further validates the use of copula function in modeling multivariate distributions. Sklar's theorem shows that, for any multivariate probability distribution, one can find a copula function that connects the joint c.d.f. with its marginal c.d.f.s. If all marginal c.d.f.s are continuous, then the copula function is unique. This theorem is important because it provides the theoretical foundation for the application of copulas and paves the way that leads to copula's current popularity. We give Sklar's theorem in the following according to Nelsen (2006):

Theorem 4.2 (Sklar's theorem)**.** *If $X_1, ..., X_n$ are random variables defined on a common probability space, with one-dimensional c.d.f.s $F_{X_i}(x_i) = P(X_i \leq x_i)$ and the joint c.d.f. $F_{X_1,...X_n}(x_1, ...x_n) = P(X_1 \leq x_1, ..., X_n \leq x_n)$, then there exists an n-dimensional copula function $C(u_1, ..., u_n)$ such that $F_{X_1,...X_n}(x_1, ...x_n) = C(F_{X_1}(x_1), ..., F_{X_n}(x_n))$ for all $x_i \in R$, $i = 1, ..., n$. If the univariate marginal c.d.f.s, $F_{X_1}, ..., F_{X_n}$, are all continuous, then the copula function is unique and can be obtained via the inversion method:*

$$C(u_1, ..., u_n) = F_{X_1,...X_n}(F_{X_1}^{-1}(u_1), ..., F_{X_n}^{-1}(u_n))$$

where $F_{X_i}^{-1}(u_i) = \inf\{x : F_{X_i}(x_i) \geq u_i\}$. Otherwise, the copula is uniquely determined at points $u_1, ...u_n$, where u_i is in the range of $F_i, i = 1, ..., n$.

Sklar's theorem allows researchers to reconstruct a multivariate distribution through modeling two parts separately: the univariate marginal distributions and the copula. This reduces the complexity of modeling a multivariate distribution.

One feature that is frequently discussed in copula functions is the tail dependence. This is a special non-linear dependence measure that quantifies the amount of dependence between extreme values of two random variables. Nelsen (2006) gives the definition of tail dependence in the following way.

Definition 4.3. *Let X and Y be continuous variables with distribution functions F and G, respectively. The* upper tail dependence parameter λ_U *and the* lower tail dependence parameter λ_L *are the limits (if they exist) defined as follows*

$$\begin{aligned} \lambda_U &= \lim_{t \to 1^-} P\left[Y > G^{-1}(t) | X > F^{-1}(t)\right] \\ &= \lim_{t \to 1^-} \frac{1 - 2t - C(t,t)}{1 - t} \end{aligned} \tag{4.1.2}$$

$$\begin{aligned} \lambda_L &= \lim_{t \to 0^+} P\left[Y \leq G^{-1}(t) | X \leq F^{-1}(t)\right] \\ &= \lim_{t \to 0^+} \frac{C(t,t)}{t} \end{aligned} \tag{4.1.3}$$

The conditional probabilities in Equations 4.1.2 and 4.1.3 can be written as

$$\frac{\text{Probability that both } X \text{ and } Y \text{ go to extreme values}}{\text{Probability that } X \text{ goes to extreme values}}$$

Intuitively, if X and Y are highly tail dependent, then if X goes to extreme values, no matter how extreme it is, the probability that Y goes to the same extent of extremity is almost always high. Therefore it is possible that two random variables are moderately dependent in the

normal range, but are highly tail dependent when it comes to extreme values. This concept is especially useful in applications such as finance, since stocks may tend to go to extremely high or low values at the same time. In addition, tail dependence only depends on the copula function of two random variables, which can be seen in Equation 4.1.2 and 4.1.3. This is why tail dependence is commonly discussed in copula functions. In the next Section, we will discuss several parametric copulas and their tail dependence will also be presented.

4.2 Parametric Copula

Since the introduction of copula theory (Sklar 1959), a number of legitimate parametric copula functions have been found and applied to real-world problems. Most commonly discussed are two types of parametric families: the elliptical copula family and the Archimedean copula family.

4.2.1 Elliptical Copula

In this Section, we will briefly introduce the two most popular elliptical copulas in the literature: the Gaussian copula and the t copula. For a more general description of the elliptical copula, refer to Cambanis et al. (1981), Fang et al. (1987), Embrechts et al. (2003).

Gaussian Copula

The Gaussian copula, as the name indicates, is the copula corresponding to the Gaussian distribution. Therefore, using the inversion method in Sklar's theorem, the Gaussian copula

can simply be defined as

$$C^{Gauss}(u_1, u_2, ..., u_n) = \Phi_R^n(\Phi^{-1}(u_1), ..., \Phi^{-1}(u_n))$$

where Φ_R^n is the c.d.f. for a n-variate standard Gaussian distribution with correlation matrix R, and Φ is the c.d.f. for a univariate standard Gaussian distribution. In the bivariate case, the Gaussian copula can be written as

$$C^{Gauss}(u, v) = \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi(1-\rho^2)^{1/2}} \exp - \frac{x_1^2 - 2\rho x_1 x_2 + x_2^2}{2(1-\rho^2)} dx_1 dx_2$$

where ρ is the correlation coefficient between the two original random variables.

t Copula

Similar to the Gaussian copula, the t copula can be constructed using an n-variate t-distribution as follows

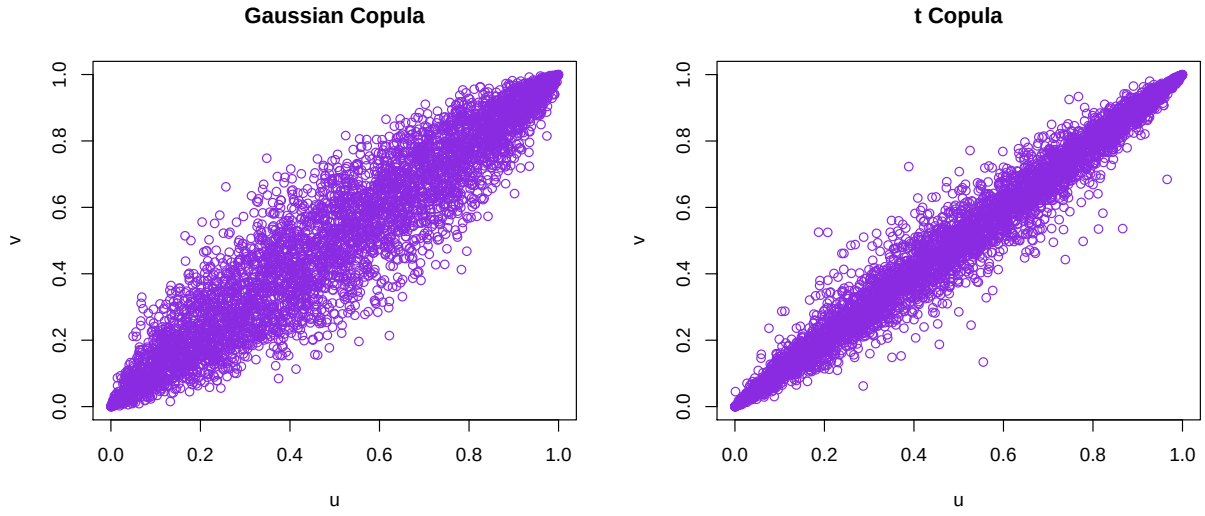
$$C^T(u_1, u_2, ..., u_n) = t_{\nu,\mathbf{R}}^n(t_\nu^{-1}(u_1), ..., t_\nu^{-1}(u_n))$$

where $t_{\nu,\mathbf{R}}^n$ is the c.d.f for a n-variate t distribution and t_ν is the c.d.f. of the univariate marginals. The multivariate t random variable can be derived as $T = \sqrt{v}\mathbf{Y}/\sqrt{S}$, where $S \sim \chi_\nu^2$ and $\mathbf{Y} \sim N(\mathbf{0}, \mathbf{R})$ (a multivariate standard normal) are independent. In the bivariate case, the expression for the t copula is

$$C^T(u, v) = \int_{-\infty}^{t_\nu^{-1}(u)} \int_{-\infty}^{t_\nu^{-1}(v)} \frac{1}{2\pi(1-\rho^2)^{1/2}} \{1 + \frac{x_1^2 - 2\rho x_1 x_2 + x_2^2}{\nu(1-\rho^2)}\}^{-(\nu+2)/2} dx_1 dx_2 \qquad (4.2.1)$$

where ρ is the correlation coefficient of the two original random variables.

In Figure 4.1, we plot 5000 random numbers generated from a bivariate Gaussian and a t copula with the same degree of dependence (Spearman's ρ=0.9 or Pearson correlation =0.9079). We can see that both plots are symmetric, but in the corners the t copula tends to be narrower and denser, indicating higher dependence. This is because the t copula exhibits tail dependence, while the Gaussian copula does not. The Gaussian copula has tail dependence only when the Pearson correlation between the two random variables equals to 1, which is rarely true. More details about these including the theoretical proofs can be found in Embrechts et al. (2003).



(a) Symmetric but no tail dependence (b) Upper and lower tail dependence

Figure 4.1: 5000 random numbers from the Gaussian copula and the t copula

4.2.2 Archimedean Copula

Most of the early legitimate copulas are for bivariate cases because it was quite difficult for early researchers to find proper multivariate copulas with explicit expressions. The invention of the Archimedean copula provides an easy way to extend bivariate copulas to multivariate

cases. The Archimedean copula originally appeared not in statistics, but in the study of probabilistic metric spaces. For an account of this history, see Schweizer (1991) and the references cited therein.

According to Nelsen (2006), a function, $C : [0,1]^n \rightarrow [0,1]$ is called an n-dimensional Archimedean copula if it can be represented through a special generator function ψ. That is,

$$C(u_1, ..., u_n; \theta) = \psi^{[-1]}(\psi(u_1; \theta) + ... + \psi(u_n; \theta)) \tag{4.2.2}$$

where θ is the copula parameter, and $\psi : [0,1] \rightarrow [0,\infty]$ is a function that is continuous, strictly decreasing, and convex. Also, $\psi(1) = 0$, and its pseudo-inverse, $\psi^{[-1]}$, is defined as

$$\psi^{[-1]}(t) = \begin{cases} \psi^{-1}(t), & 0 \leq t \leq \psi(0) \\ 0, & 0 \leq \psi(0) \leq \infty \end{cases}$$

In particular, if $\psi(0) = \infty$, then $\psi^{[-1]} = \psi^{-1}$ and ψ is called a strict generator. Some researchers represent the Archimedean copulas by ψ^{-1} instead of ψ.

Many Archimedean copulas have only one parameter that is often related to dependence measures among the original random variables. For example, in the bivariate case, a two-dimensional copula proposed by Clayton (1978) has the form

$$C(u, v; \theta) = [\max(u^{-\theta} + v^{-\theta} - 1, 0)]^{-1/\theta}$$

where the parameter, θ, can be expressed analytically as a function of the Kendall's τ (Kendall,1938) which is a commonly used non-linear dependence measure:

$$\tau = \frac{\theta}{\theta + 2}$$

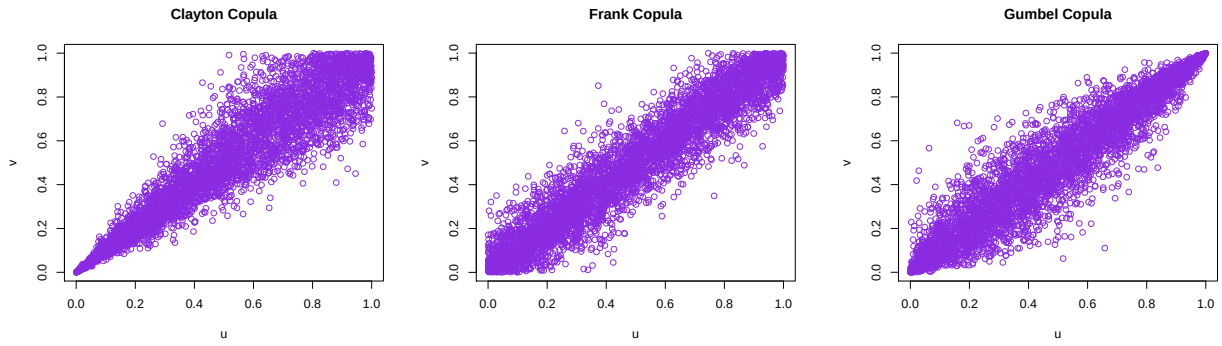
It can be easily shown that the Clayton copula is a two-dimensional Archimedean copula with the generator function $\psi(t;\theta) = \frac{1}{\theta}(t^{-\theta} - 1)$. It can be extended to higher dimensions following Equation 4.2.2. The simplicity of generalizing to higher dimensions makes the Archimedean copulas very attractive. Three commonly used Archimedean copulas and their generator functions are given in Table 4.1.

Table 4.1: Three Archimedean Copula Families

Family	θ	$\psi(t)$	Ref. as cited in Nelsen (2006)
Clayton	$[-1, \infty] \setminus \{0\}$	$\frac{1}{\theta}(t^{-\theta} - 1)$	Clayton (1978)
Frank	$[-\infty, \infty] \setminus \{0\}$	$-\log \frac{e^{-\theta t} - 1}{e^{-\theta} - 1}$	Frank (1979)
Gumbel	$[1, \infty]$	$(-\log t)^{\theta}$	Gumbel (1960)

In Figure 4.2, we plot 5000 random numbers generated from each Archimedean copula in Table 4.1. All three copulas have the same Kendall's τ of 0.9. From the Figure, we can see that, first of all, the Frank copula is symmetric and the other two are not. Considering the behavior in the corners, it is obvious that the Clayton copula displays higher dependence in the lower-left corner and the Gumbel copula displays higher dependence in the upper-right corner. This is because the Gumbel copula exhibits upper tail dependence and the Clayton copula displays lower tail dependence. The Frank copula does not exhibit any tail dependence.

However, the Archimedean copula has one obvious limitation: the dependence structure among any sub-dimensions has to be identical. For example, in three dimensional cases, the formula for the copula function is $C(u_1, u_2, u_3; \theta) = \psi^{[-1]}(\psi(u_1;\theta) + \psi(u_2;\theta) + \psi(u_3;\theta))$. Then if we want to examine the dependence between u_1 and u_2, we can just let $u_3 = 1$ and get $C(u_1, u_2; \theta) = \psi^{[-1]}(\psi(u_1;\theta) + \psi(u_2;\theta))$. However, no matter which two dimensions we take, the resulting two dimensional copula will have exactly the same form. But in bivariate



(a) Lower tail dependence (b) No tail dependence (c) Upper tail dependence

Figure 4.2: 5000 random numbers from three Archimedean copulas

cases, the Archimedean copula family is still one of the most popular options. See Embrechts et al. (2003) and Nelsen (2006) for more details about Archimedean copulas.

4.3 Non-Parametric Copula

As discussed in the last Section, each parametric copula is useful only for certain cases, e.g., Gumbel copula for upper tail dependence, and Clayton copula for lower tail dependence. One then has to select the appropriate parametric copula based on the data. This selection can be done by trying a list of parametric copulas and select the one giving the highest likelihood value. However, it can be computationally expensive if one tries too many parametric copulas, and choosing which copulas to try is a subjective decision which may need prior knowledge. Therefore it is natural to consider non-parametric copulas as an alternative. While there are many non-parametric methods available, we will mainly focus on the empirical copula and the kernel-based copula here.

4.3.1 Empirical Copula

According to Chen and Huang (2007) and Omelka et al. (2009), the proposal of empirical copula is due to Deheuvels (1979). Given a set of i.i.d. bivariate data $\{(X_i, Y_i)\}_{i=1}^n$, the empirical copula estimator has the following form:

$$C_n(u, v) = \frac{1}{n} \sum_{i=1}^{n} I\{\hat{U}_i \leq u, \hat{V}_i \leq v\} \tag{4.3.1}$$

where $\hat{U}_i = F_n(X_i)$, $\hat{V}_i = G_n(Y_i)$, F_n and G_n are the empirical cumulative distribution functions of the marginals, and $I\{S\}$ is the indicator function of set S. Weak convergence studies of this estimator can be found in Gänssler and Stute (1987), Fermanian et al. (2004), Tsukahara (2005). Omelka et al. (2009) suggest using an asymptotically equivalent modified version of the empirical copula as follows

$$C_n^{ME}(u, v) = \frac{1}{n} \sum_{i=1}^{n} I\{\hat{U_i}^{ME} \leq u, \hat{V_i}^{ME} \leq v\} \tag{4.3.2}$$

where $\hat{U_i}^{ME} = \frac{n}{n+1} F_n(X_i), \hat{V_i}^{ME} = \frac{n}{n+1} G_n(Y_i)$. This adjusted version shifts the original U_i and V_i a bit closer to the left corner of the unit interval $[0, 1]$, and also avoid having exactly ones and zeros which could potentially cause numerical problems. Omelka et al. (2009) claims that in their Monte Carlo simulation the adjusted empirical copula estimator in Equation 4.3.2 has better estimates for the copula function than the unadjusted one in Equation 4.3.1. A smoothed version of the empirical copula was proposed by Fermanian et al. (2004). The smoothed empirical estimator uses an idea similar to the kernel-based estimation, and has the following form

$$\hat{C}_n^{(SE)}(u, v) = \hat{H}_n(\hat{F}_n^{-1}(u), \hat{G}_n^{-1}(v))$$

where

$$\hat{H}_n(x,y) = \frac{1}{n}\sum_{i=1}^{n} K_n(x - X_i, y - Y_i)$$

$$\hat{F}_n(x) = \lim_{y\to+\infty} \hat{H}_n(x,y)$$

$$\hat{G}_n(y) = \lim_{x\to-\infty} \hat{H}_n(x,y)$$

$$K_n(x,y) = K(\frac{x}{b_n}, \frac{y}{b_n})$$

$$K(x,y) = \int_{-\infty}^{x}\int_{-\infty}^{y} k(s,t)dsdt$$

where $k(s,t)$ is a given bivariate kernel density function, and b_n is a bandwidth sequence with a limit of zero when n tends to ∞.

4.3.2 Kernel-Based Copula

Another popular class of copula estimators uses the idea of kernel estimation for general distribution functions and densities. But the copula function is different from a general distribution function in that it has a bounded support of $[0,1]^n$. This bounded support is not accounted for by the regular kernel estimator. Therefore directly using the kernel estimator for copulas leads to the so-called boundary bias, meaning that the estimation at the boundary of the support can be very biased. Moreover, some copula functions are unbounded in the boundary area. For example, the Clayton copula goes to infinity in the corner of $(0,0)$. When estimating this type of copulas, their unboundedness in the boundary area can also cause boundary bias.

In order to overcome the boundary bias in estimating copulas, we introduce here a number of modified kernel copula estimators that have been proposed in the literature. We will use

these copula estimators to construct the non-parametric CHOMP in Section 5.3. We will also compare the performance of these estimators in a simulation study in Section 7.3.

Mirror-Reflection Kernel Estimator

Gijbels and Mielniczuk (1990) use a technique known as reflection/mirror-reflection in their kernel-based copula estimators. The basic idea of this approach is to expand the support by mirroring or reflection. In two dimensional cases, the support for a copula function is $[0,1]^2$. Then everything in $[0,1]^2$ is reflected eight times in eight different directions. One example is given in Figure 4.3. Combining the original support and the eight reflected

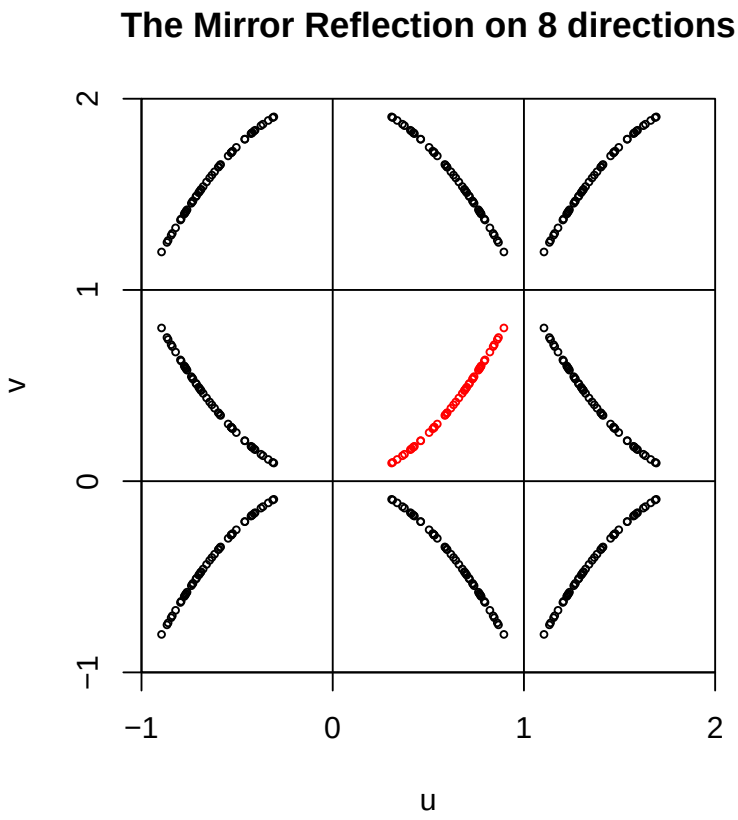


Figure 4.3: Mirror reflection
The original points are plotted in red, and the others are the reflected points in eight different directions.

supports, one has an expanded support of $[-1,2]^2$. Then the regular kernel estimator is applied in this expanded support, and the part within the original support $[0,1]^2$ is used in the estimation for the copula density. The copula can then be estimated by integrating the copula density. Mathematically, according to Omelka et al. (2009), if one uses a product kernel $k(x,y) = k(x)k(y)$, the mirror-reflection kernel estimator has the following form

$$\hat{C}_n^{(\text{mirror})}(u,v) = \frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{9}\left[K\Big(\frac{u-\hat{U}_i^{(l)}}{h_n}\Big) - K\Big(\frac{-\hat{U}_i^{(l)}}{h_n}\Big)\right] \times \left[K\Big(\frac{v-\hat{V}_i^{(l)}}{h_n}\Big) - K\Big(\frac{-\hat{V}_i^{(l)}}{h_n}\Big)\right]$$

where

$$\begin{aligned}\{(\hat{U}_i^{(l)}, \hat{V}_i^{(l)}), i = 1,...,n, l = 1,...,9\} = \{&(\hat{U}_i, \hat{V}_i), (\hat{U}_i, -\hat{V}_i), (-\hat{U}_i, \hat{V}_i), (-\hat{U}_i, -\hat{V}_i),\\ &(\hat{U}_i, 2-\hat{V}_i), (-\hat{U}_i, 2-\hat{V}_i), (2-\hat{U}_i, \hat{V}_i), (2-\hat{U}_i, -\hat{V}_i), (2-\hat{U}_i, 2-\hat{V}_i), i = 1,...,n\}\end{aligned}$$

Probit-Transformation Kernel Copula

Another natural way to overcome the boundary bias problem is through transformation of the random variables. Specifically, we know that the difficulty is caused by the bounded support. Then one might consider transforming the random variables so that the new variables have an unbounded support. Then the regular kernel density estimation can be performed on the transformed support. This idea is theoretically feasible because the copula function is invariant to transformations with increasing functions. In other words, as long as the transformation function is an increasing function, the transformed random variables will have the same copula function as the untransformed ones. See Nelsen (2006) for more details about the invariance of copula.

The estimator proposed by Fermanian, Radulovic and Wegkamp (2004) follows this transformation idea. In the two dimensional case, the support, $[0,1]^2$, of the copula estimator is

transformed back to the support of the original two random variables. This original support doesn't have the boundary bias problem. However, this estimator is criticized because of its strong dependence on the marginal distributions. Omelka et al. (2009) shows in Monte Carlo simulations that, for a given copula, the success of this estimator depends crucially on the marginals.

To overcome this problem, Omelka et al. (2009) propose the following procedure for the transformation copula estimator. First, uniform pseudo-observations are estimated by $\hat{U}_i = \frac{n}{n+1}F_n(X_i)$ and $\hat{V}_i = \frac{n}{n+1}G_n(Y_i)$. Second, new transformed random variables are obtained by $\hat{S}_i = \Psi^{-1}(\hat{U}_i)$ and $\hat{T}_i = \Psi^{-1}(\hat{V}_i)$, where Ψ is a given distribution function. Then the regular kernel-based c.d.f. estimation can be applied to $(\hat{S}_i, \hat{T}_i)$. This probit-transformation kernel copula estimator, $\hat{C}_n^{(T)}$, can be written as

$$\hat{C}_n^{(pt0)}(u,v) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{\Psi^{-1}(u) - \Psi^{-1}(\hat{U}_i)}{h_n}\right) \times K\left(\frac{\Psi^{-1}(u) - \Psi^{-1}(\hat{V}_i)}{h_n}\right) \tag{4.3.3}$$

The use of Ψ in this estimator removes the dependence on the marginals, F_n and G_n. In addition, this estimator also avoids the boundary bias problem if Ψ satisfying the condition that $\frac{\Psi'(x)^2}{\Psi(x)}$ is bounded. So far researchers have only used the normal cumulative distribution function for Ψ. The choice of Ψ is still an open question. With this transformation estimator, we still need the boundedness condition. However, now we only need boundedness in the support of $(\hat{S}_i, \hat{T}_i)$ instead of $[0,1]^2$. This new boundedness condition is considerably weaker as shown by Omelka et al. (2009) in a bivariate normal benchmark example. By using the spherical kernel instead of the product kernel in Equation 4.3.3, Charpentier et al. (2007) suggests the following probit-transformation kernel copula density estimator:

$$\hat{c}_n^{(pt1)}(u,v) = \frac{1}{n|\mathbf{H}_{ST}|^{1/2}\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))}\sum_{i=1}^{n} k\left(\mathbf{H}_{ST}^{-1/2}\begin{pmatrix}\Phi^{-1}(u) - \Phi^{-1}(\hat{U}_i)\\ \Phi^{-1}(v) - \Phi^{-1}(\hat{V}_i)\end{pmatrix}\right) \tag{4.3.4}$$

where k is a bivariate kernel function, $\mathbf{H}_{ST}$ is some symmetric positive-definite bandwidth matrix, Φ is the standard normal c.d.f., and ϕ is the standard normal density function. The bandwidth matrix is selected by the plug-in method in Duong and Hazelton (2003). The corresponding copula function can easily be obtained by integrating the kernel copula estimator in Equation 4.3.4 as follows:

$$\hat{C}_n^{(pt1)}(u,v) = \int_0^u \int_0^v \hat{c}_n^{(pt1)}(\delta_1, \delta_2)\ d\delta_1 d\delta_2 \tag{4.3.5}$$

Geenens et al. (2014) give the theoretical property of $\hat{c}_n^{(pt1)}(u,v)$. Based on its theoretical bias, they further give an amended version of the probit-transformation kernel copula estimator that, to some extent, mitigates the boundary bias caused by the unboundedness of the true copula density. We don't give it here since the bandwidth choice for the amended estimator hasn't been clearly given.

Improved Probit-Transformation Kernel Copula

Geenens et al. (2014) also propose an improved probit-transformation copula density estimator by using local likelihood methods instead of kernel density estimation after the probit-transformation. Using local likelihood with Gaussian kernels, the estimated density in the transformed domain will be locally similar to the probit function, hence improve the estimation accuracy in $[0,1]^2$ after back-transformed.

Using the same setting and notation as before, one wants to estimate the density function using the pseudo samples $\{\hat{S}_i, \hat{T}_i\}$ after probit-transformations. Let us denote this density function by f_{ST}. Geenens et al. (2014) give the local likelihood estimator defined by Loader (1996) as follows. Around a particular point $(s,t) \in R^2$, $\log f_{ST}$ is assumed to be well approximated by a polynomial of some order p. Classically, only local log-linear ($p = 1$) and

local log-quadratic ($p = 2$) estimators are considered. Specifically, in the first case ($p = 1$), it is assumed that, for $(\check{s}; \check{t})$ close to (s, t),

$$\log f_{ST}(\check{s}, \check{t}) \approx a_{1,0}(s,t) + a_{1,1}(s,t)(\check{s} - s) + a_{1,2}(s,t)(\check{t} - t) \triangleq P_{\mathbf{a}_1}(\check{s} - s, \check{t} - t)$$

and in the second case ($p = 2$)

$$\begin{aligned}\log f_{ST}(\check{s}, \check{t}) \approx & a_{2,0}(s,t) + a_{2,1}(s,t)(\check{s} - s) + a_{2,2}(s,t)(\check{t} - t) + a_{2,3}(s,t)(\check{s} - s)^2 \\ & + a_{2,4}(s,t)(\check{t} - t)^2 + a_{2,5}(s,t)(\check{s} - s)(\check{t} - t) \\ \triangleq & P_{\mathbf{a}_2}(\check{s} - s, \check{t} - t)\end{aligned}$$

where the two vectors

$$\begin{aligned}\mathbf{a}_1(s,t) &\triangleq (a_{1,0}(s,t), a_{1,1}(s,t), a_{1,2}(s,t)) \\ \mathbf{a}_1(s,t) &\triangleq (a_{1,0}(s,t), a_{1,1}(s,t), a_{1,2}(s,t))\end{aligned}$$

are then estimated by solving a weighted maximum likelihood problem. For $p = 1, 2$,

$$\begin{aligned}\hat{\mathbf{a}}_p(s,t) = arg\max_{\mathbf{a}_p} \Big\{ \sum_{i=1}^{n} k\left(\mathbf{H}_{ST}^{-1/2} \begin{pmatrix} s-\hat{S}_i \\ t-\hat{T}_i \end{pmatrix}\right) P_{\mathbf{a}_p}(\hat{S}_i - s, \hat{T}_i - t) \\ - n \int\int_{R^2} k\left(\mathbf{H}_{ST}^{-1/2} \begin{pmatrix} s-\check{s} \\ t-\check{t} \end{pmatrix}\right) exp(P_{\mathbf{a}_p}(\check{s} - s, \check{t} - t)) d\check{s} d\check{t} \Big\}\end{aligned}$$

where k is a bivariate kernel function and $\mathbf{H}_{ST}$ is a symmetric positive definite bandwidth matrix. Then the estimate of f_{ST} at (s, t) is $\hat{f}_{ST}^{(1)}(s,t) = exp(\hat{a}_{1,0}(s,t))$ for local log-linear, and $\hat{f}_{ST}^{(2)}(s,t) = exp(\hat{a}_{2,0}(s,t))$ for local log-quadratic. Finally, the improved probit-

transformation kernel copula density estimators for $c(u, v)$ are given as

$$\hat{c}^{ptloc(p)}(u, v) = \frac{\hat{f}_{ST}^{(p)}(\Phi^{-1}(u), \Phi^{-1}(v))}{\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))} \tag{4.3.6}$$

Geenens et al. (2014) also suggested a bandwidth selection method and provided the asymptotic properties for $\hat{c}^{ptloc(p)}(u, v)$. The corresponding copula function can be written as

$$\hat{C}^{ptloc(p)}(u, v) = \int_0^u \int_0^v \hat{c}^{ptloc(p)}(\delta_1, \delta_2)\ d\delta_1 d\delta_2 \tag{4.3.7}$$

where $\hat{c}^{ptloc(p)}$ is given in Equation 4.3.6.

Boundary Kernel Copula Estimator

Another way to get over the boundary problem is to use a boundary kernel in the kernel-based estimator. Since the support of copulas or copula densities is the unit hyper-rectangle, a suitable kernel function is the univariate boundary beta kernel of Chen (1999) with unit interval support. Following the notation in Duong (2014), the boundary beta kernel is, for $0 \leq x,\ u \geq 1$,

$$k_{u,h}^{bk}(x) = \begin{cases} beta(x; \rho(u, h), (1-u)/h^2),\ u \in [0, 2h) \\ beta(x; u/h^2, (1-u)/h^2),\ u \in [2h^2, 1-2h^2] \\ beta(x; u/h^2, \rho(1-u, h)),\ u \in (1-2h^2, 1] \end{cases}$$

where $beta(\cdot; \alpha_1, \alpha_2)$ is the density function for a beta random variable with shape parameter α_1 and α_2, and $\rho(u, h) = 2h^4 + 5/2 - (4h^4 + 6h^2 + 9/4 - u^2 - u/h^2)^{1/2}$. From the expression of this kernel, we can see that it is adjusting the bandwidth implicitly by adjusting the shape parameters of the beta function. This unique feature of the beta kernel can therefore

mitigate the boundary bias problem. The resulting univariate kernel density estimate of $f(x) : [0,1] \to [0,1]$ is

$$\hat{f}_h^{bk}(x) = \frac{1}{n}\sum_{i=1}^{n} k_{x,h}^{bk}(x_i)$$

Chen (1999) shows that, in the univariate case, the bias of the beta kernel estimator, $\hat{f}_h^{bk}(x)$, is $O(h^2)$ in the entire $[0,1]$, no matter it is in the interior or boundary of $[0,1]$. Therefore this estimator does not have the boundary bias problem. Chen (1999) also shows that the variance of the estimator in boundary regions of $[0,1]$ is negligible, compared to the variance in the interior of $[0,1]$. Lee (1996) and Olkin and Liu (2003) give the multivariate beta kernel estimator based on the work of Chen (1999). However, Duong (2014) claims it is not clear whether the multivariate beta kernel estimator still can avoid the boundary bias problem as its univariate counterpart does. So instead of using the multivariate beta kernel, Duong (2014) uses the product beta kernel which is the product of two univariate beta kernel in the bivariate case. This leads to the following estimator

$$\hat{f}_{h_1,h_2}^{bk}(x,y) = \frac{1}{n}\sum_{i=1}^{n} k_{x,h_1}^{bk}(x_i) \times k_{y,h_2}^{bk}(y_i)$$

The use of product beta kernel limits the performance of this estimator, so Duong (2014) proposes a hybrid estimator using the spherical symmetric kernels in the interior region of $[0,1]^2$, whilst using the product beta kernel in the boundary region. Therefore the spherical symmetric kernels induce a performance gain over the product beta kernel in the interior of $[0,1]^2$, and the product beta kernel overcomes the boundary bias problem in the boundary regions of $[0,1]$. The new hybrid kernel estimator has the following form

$$\hat{f}_{\mathbf{H}}^{hbk}(x,y) = \frac{1}{n}\sum_{i=1}^{n}[k_{x,h_1}^{bk}(x_i)k_{y,h_2}^{bk}(y_i)I\{(x_i,y_i) \in \bar{\Omega}\} + k_{\mathbf{H}}^{ssk}(x-x_i, y-y_i)I\{(x_i,y_i) \in \Omega\}]$$

where $k_{\mathbf{H}}^{ssk}$ is a regular spherical symmetric kernel with $\mathbf{H}$ as the bandwidth matrix, $\Omega = [\eta_1, 1-\eta_1] \times [\eta_2, 2-\eta_2]$, where η_j is the j-th diagonal of $\mathbf{H}$. Duong (2014) also gives a data-driven bandwidth selection method to get $\mathbf{H}$, h_1 and h_2 sequentially. Then the resulting hybrid kernel copula density estimator is

$$\hat{c}_{\mathbf{H}}^{hbk}(u,v) = \frac{1}{n}\sum_{i=1}^{n}[k_{u,h_1}^{bk}(u_i)k_{v,h_2}^{bk}(v_i)I\{(u_i,v_i) \in \bar{\Omega}\} + k_{\mathbf{H}}^{ssk}(u-u_i, v-v_i)I\{(u_i,v_i) \in \Omega\}] \quad (4.3.8)$$

where $k_{\mathbf{H}}^{ssk}$ is a regular spherical symmetric kernel with $\mathbf{H}$ as the bandwidth matrix, $\Omega = [\eta_1, 1-\eta_1] \times [\eta_2, 2-\eta_2]$, where η_j is the j-th diagonal of $\mathbf{H}$. The corresponding copula function is then estimated as

$$\hat{C}_{\mathbf{H}}^{hbk}(u,v) = \int_0^u \int_0^v \hat{c}_{\mathbf{H}}^{hbk}(\delta_1, \delta_@)\ d\delta_1 d\delta_2 \quad (4.3.9)$$

where $\hat{c}_{\mathbf{H}}^{hbk}$ is given in Equation 4.3.8

In addition to the empirical copula and kernel-based copula, other non-parametric methods are also proposed in the literature, including Fourier methods (Lowin 2010), Bernstein polynomials (Li et al. 1998; Sancetta and Satchell 2004; Hurd et al. 2005), and wavelet methods (Genest et al. 2009; Autin et al. 2010; Morettin et al. 2011).

In summary, the copula is a very useful tool for describing the dependence structure among random variables. It captures the intrinsic dependence information among random variables regardless of their marginal distributions. In other words, two sets of variables, $\{X_1, Y_1\}$ and $\{X_2, Y_2\}$, can have a common copula function, $C(u,v)$, but two different sets of marginal distributions, $\{F_1(x), G_1(y)\}$ and $\{F_2(x), G_2(y)\}$. In this Chapter, we briefly introduce the copula theory, and have discussed the properties of several important copula families. To overcome the model mis-specification problem for parametric copulas, we introduce several non-parametric copulas which can handle various dependence structures (linear or non-linear) among the random variables. In the next Chapter, we will introduce the CHOMP

model that is used in our proposed time series clustering algorithm. The CHOMP model uses the copula to characterize the temporal dependence in time series. In fact, the advantage of CHOMP over other models is largely due to its use of the copula, since the copula allows the CHOMP to model both the strength and structure of the temporal dependence in real-valued time series. We will also see how the use of copulas influence the practical behaviors of time series generated by CHOMP.

Chapter 5

Copula-Based Higher Order Markov Process

As mentioned in Section 3.3, we use a new time series model called CHOMP in our proposed time series clustering algorithm because it is able to extract more complete information from the data. In Chapter 4, we introduce the concept of copula and its advantages in modeling dependence. In this Chapter, we give more details about the CHOMP model. We will explain how it incorporates the concept of copula and why it can capture more complete information from time series data.

In Section 5.1 of this Chapter, we introduce the general concept of a higher order Markov process, and give a more general form of the results in Ibragimov (2009). In Section 5.2, we first give the copula-based characterization of the higher order Markov process by Ibragimov (2009). As Ibragimov's results only apply to stationary time series, we give one of our main results in Corollary 5.7, which relaxes the stationarity assumption. This makes the CHOMP able to model non-stationary time series, and capture their arbitrary profile information.

Finally, in Section 5.3, we give details on statistical inference for CHOMP, including model estimation, likelihood calculation, and how to choose the copula function. Unless otherwise stated, all the Markov processes discussed in this work are assumed to have discrete time points and continuous state spaces.

5.1 Higher Order Markov Process

In stochastic processes, the commonly used Markov property is actually the first order Markov property. Specifically, given a discrete time stochastic process $\{X_t\}_{t=1}^{\infty}$ with real state space R, where X_t can be any continuous random variables on R, the (first order) Markov property states that, for any $x_1, ..., x_{n+1} \in R$,

$$P(X_{n+1} \leq x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, ..., X_1 = x_1) = P(X_{n+1} \leq x_{n+1} | X_n = x_n)$$

In other words, given the realization of the previous random variable, the probabilistic behavior of the current random variable is independent of all other past random variables. The first order Markov property can be naturally extended to higher orders. The Markov property with order q can be stated in the following way: for any $x_1, ..., x_{n+1} \in R$,

$$P(X_{n+1} \leq x_{n+1} | X_n = x_n, ..., X_{n-q+1} = x_{n-q+1}, ..., X_1 = x_1) = P(X_{n+1} \leq x_{n+1} | X_n = x_n, ..., X_{n-q+1} = x_{n-q+1}) \quad (5.1.1)$$

Any stochastic process with the property in Equation 5.1.1 is then called a Markov process of order q. If $q \geq 2$, we call it a higher order Markov process (HOMP).

We define a $\star$ product for general c.d.f.s following the $\star$ product defined in Ibragimov (2009) for copulas. Let F and G be m and n-dimensional c.d.f.s on $\bar{R}^m$ and $\bar{R}^n$ such that for any $\gamma_1, ..., \gamma_b \in \bar{R} = [-\infty, \infty]$

$$\begin{aligned} F(x_1 = \infty, ..., x_{m-b} = \infty, \gamma_1, ..., \gamma_b) &= G(\gamma_1, ..., \gamma_b, y_1 = \infty, ..., y_{n-b} = \infty) \\ &= H(\gamma_1, ..., \gamma_b), \end{aligned} \tag{5.1.2}$$

where $m, n > b \geq 1$, and H is a b-dimensional c.d.f. on $\bar{R}^b$. In other words, the c.d.f. for the last b dimensions of F is the same as that for the first b dimensions of G.

Definition 5.1 ($\star^b$-product)**.** *Let F, G and H be defined as above, an $(m+n-b)$-dimensional distribution function W is defined as $F \star^b G$ if*

$$\begin{aligned} W(z_1, ..., z_{m+n-b}) = \int ... \int_S & F_{1,...,m|m-b+1,...,m}(x_1, ..., x_{m-b}, \gamma_1, ..., \gamma_b) \\ & \cdot G_{1,...,n|1,...,b}(\gamma_1, ..., \gamma_b, y_1, ..., y_{n-b}) \cdot h(\gamma_1, ..., \gamma_b) \; d\gamma_1 ... d\gamma_b \end{aligned}$$

where

$$\begin{aligned} &F_{1,...,m|m-b+1,...,m}(x_1, ..., x_{m-b}, \gamma_1, ..., \gamma_b) = \frac{\partial^b F(x_1, ..., x_{m-b}, \gamma_1, ..., \gamma_b)}{\partial\gamma_1 ... \partial\gamma_b} \Big/ \frac{\partial^b H(\gamma_1, ..., \gamma_b)}{\partial\gamma_1 ... \partial\gamma_b}, \\ &G_{1,...,n|1,...,b}(\gamma_1, ..., \gamma_b, y_1, ..., y_{n-b}) = \frac{\partial^b G(\gamma_1, ..., \gamma_b, y_1, ..., y_{n-b})}{\partial\gamma_1 ... \partial\gamma_b} \Big/ \frac{\partial^b H(\gamma_1, ..., \gamma_b)}{\partial\gamma_1 ... \partial\gamma_b}, \\ &h(\gamma_1, ..., \gamma_b) = \frac{\partial^b H(\gamma_1, ..., \gamma_b)}{\partial\gamma_1 ... \partial\gamma_b}. \end{aligned}$$

Then based on the higher order Markov property and Bayes Theorem, we give the following results in Theorem 5.2.

Theorem 5.2. *A real valued stochastic process $\{X_t\}_{t\in T}$ is a Markov process of order $q, q \geq 1$, if and only if, for all $t_i \in T, i = 1, ..., n$, such that $t_1 < ... < t_n$ and $n \geq q + 1$,*

$$F(x_{t_1}, ..., x_{t_n}) = F_{t_1,...,t_{q+1}} \star^q ... \star^q F_{t_{n-q},...t_n}$$

Adding a stationarity condition (Definition 5.3) to Theorem 5.2 yields Corollary 5.4.

Definition 5.3. *A stochastic process $\{X_t\}_{t\in T}$ is called stationary if, for all l and τ, and for all $t_1, ..., t_l$,*

$$F(x_{t_1+\tau}, ..., x_{t_l+\tau}) = F(x_{t_1}, ..., x_{t_l})$$

where $F(x_{t_1+\tau}, ..., x_{t_l+\tau})$ represents for the c.d.f. of $\{X_{t_1+\tau}, ..., X_{t_l+\tau}\}$.

Corollary 5.4. *A real valued stochastic process $\{X_t\}_{t=1}^{\infty}$ is a stationary Markov process of order q, $q \geq 1$, if and only if, for all $n \geq q + 1$,*

$$F_{1,...,n}(x_1, ..., x_n) = \underbrace{F \star^q F \star^q ... \star^q F}_{n-q+1}(x_1, ..., x_n)$$

where F is a $(q + 1)$-marginal distribution functions such that $F_{i_1,...,i_s} = F_{i_1+h,...,i_s+h}$, $0 \leq h \leq q + 1 - i_s$, $1 \leq i_1 \leq ... \leq i_s \leq q + 1$, $s = 2, ..., q$, and $F_{i_1,...,i_s}$ denotes the s-dimensional marginal of F, that is, $F_{i_1,...,i_s} = F(x_1, ..., x_{q+1})|_{\{x_i=\infty,\ i\neq i_1,...,i_s\}}$.

Thus Theorem 5.2 shows that a real valued Markov process with order q can be fully determined by $(n-q)$ $(q+1)$-dimensional distributions. Throughout the rest of this work, we refer to these distributions as "deterministic distributions" or "D-distributions". If the process is stationary, the D-distributions become identical and the process can be specified by only one $(q + 1)$-dimensional distribution. Therefore modeling a stationary Markov process of order q now amounts to modeling a $(q + 1)$-dimensional distribution. In Section 5.2 we will give

extensions to non-stationary cases with the help of copula theory.

Now we give the transition c.d.f. for the Markov process with order q. The transition c.d.f. is the counterpart of the transition probability matrix of a Markov process with discrete state space. Let, as before, $\{X_t\}_{t\in T}$ be a Markov process of order q. Then for any $t_1 < ... < t_{q+1} \in T$, $x_{t_1}, ..., x_{t_{q+1}} \in R$, and all Borel sets $A \in \mathcal{B}(R)$ of the form $(-\infty, x_{q+1})$, the transition c.d.f. for a Markov process with order q is

$$\begin{aligned} &P(X_{t_{q+1}} \in A | X_{t_1} = x_{t_1}, ...X_{t_q} = x_{t_q}) \\ &\qquad = \frac{\partial^q F_{t_1,...,t_q,t_{q+1}}(x_{t_1}, ..., x_{t_q}, x_{t_{q+1}})}{\partial x_{t_1}...\partial x_{t_q}} \Big/ \frac{\partial^q F_{t_1,...,t_q}(x_{t_1}, ..., x_{t_q})}{\partial x_{t_1}...\partial x_{t_q}}, \end{aligned} \tag{5.1.3}$$

where $F_{t_1,...,t_q,t_{q+1}}(x_1, ..., x_q, x_{q+1})$ is the corresponding D-distribution for dimension (t_1, ... , t_{q+1}) and $F_{t_1,...,t_q}(x_1, ..., x_q)$ is its marginal of the first q dimensions.

Further, given a sequence of values $(x_1, ..., x_n) \in R$, the likelihood of this sequence can be easily derived using the Chapman-Kolmogorov equation. Let $f(x_1, ..., x_n)$ be the likelihood that $(x_1, ..., x_n)$ is generated by the q-th order Markov process whose transition c.d.f.s are denoted as in Equation 5.1.3, then one has

$$\begin{aligned} f(x_1, ..., x_n) =& f(x_n | x_{n-q}, ..., x_{n-1}) \cdot ... \cdot f(x_{q+1} | x_1, ..., x_q) \\ & \cdot f(x_1, \ ..., \ x_q) \end{aligned} \tag{5.1.4}$$

where $f(x_m | x_{m-q}, ..., x_{m-1}) = \partial P(X_m \leq x_m | X_{m-q} = x_{m-q}, ...X_{m-1} = x_{m-1}) \ / \ \partial x_m$, $m = q+1, ..., n$ and $f(x_1, ..., x_q) = \partial^q F_{1,...,q}(x_1, ..., x_q)/\partial x_1...\partial x_q$.

The result given in Section 5.1 is, in fact, a simple generalization of the work by Ibragimov (2009). We will give Ibragimov's work in Section 5.2. By comparing the two, we will see that Ibragimov's results in Theorem 5.5 and Corollary 5.6 are in the form of copula

functions, whereas the results here in Theorem 5.2 and Corollary 5.4 are in the form of general distribution functions. In other words, Ibragimov's work tries to model the D-distribution using copulas. However, with the generalization, one has a more general framework so that techniques other than copulas can also be applied to estimate the D-distributions.

5.2 Copula-Based Higher Order Markov Process

In Section 5.2, we give the copula-based characterization of the HOMP by Ibragimov (2009). As mentioned in Section 5.1, a HOMP can be fully determined by a group of D-distributions or just one D-distribution under stationarity. Ibragimov (2009) uses the copula to model these D-distributions, resulting in the copula-based higher order Markov process (CHOMP). We give Ibragimov's results in Theorem 5.5 and Corollary 5.6.

5.2.1 Ibragimov's CHOMP

Theorem 5.5. *A real valued stochastic process* $\{X_t\}_{t\in T}$ *is a Markov process of order* q, $q \geq 1$, *if and only if, for all* $t_i \in T, i = 1, ..., n$, $n \geq q+1$ *such that* $t_1 < ... < t_n$,

$$C_{t_1,...,t_n} = C_{t_1,...,t_{q+1}} \star^q ... \star^q C_{t_{n-q},...,t_n}$$

By comparing Theorem 5.5 with Theorem 5.2, we see that the c.d.f. in Theorem 5.2, $F(x_{t_1}, ..., x_{t_n})$, and the D-distributions are now characterized by a set of copula functions

in the following form:

$$F(x_{t_1}, ..., x_{t_n}) = C_{t_1,...,t_n}(u_{t_1}, ..., u_{t_n})$$
$$F(x_{t_1}, ..., x_{t_{q+1}}) = C_{t_1,...,t_{q+1}}(u_{t_1}, ..., u_{t_{q+1}})$$
$$...$$
$$F(x_{t_{n-q}}, ..., x_{t_n}) = C_{t_{n-q},...,t_n}(u_{t_{n-q}}, ..., u_{t_n})$$

where $u_{t_i} = F(x_{t_i})$ stands for the t_ith marginal.

In Theorem 5.5, a q-th order Markov process can be characterized by a set of $(q+1)$-dimensional copula functions and a set of univariate marginal distributions. We denote this copula-based higher order Markov process by CHOMP(q). However, the CHOMP model characterized in Theorem 5.5 is not very useful practically because estimating that many multivariate copula functions is extremely difficult. To make it practically more useful, the stationarity condition is added to Theorem 5.5, which yields Corollary 5.6.

Corollary 5.6. *A sequence of identically distributed random variables $\{X_t\}_{t=1}^{\infty}$ is a stationary Markov process of order q, $q \geq 1$, if and only if for all $n \geq q+1$,*

$$C_{1,...,n}(u_1, ..., u_n) = \underbrace{C \star^q C \star^q ... \star^q C}_{n-q+1}(u_1, ..., u_n)$$

where C is a $(q+1)$-dimensional copula such that $C_{i_1,...,i_l} = C_{i_1+h,...,i_l+h}$, $0 \leq h \leq q+1-i_l$, $1 \leq i_1 \leq ... \leq i_l \leq q+1$, $l = 2, ..., q$, and $C_{j_1,...,j_l}$, $1 \leq j_1 \leq ... \leq j_l \leq q+1$, denotes the corresponding marginal of C : $C_{j_1,...,j_l} = C|_{u_i=1, i \neq j_1,...,j_l}$.

In Corollary 5.6, all the copula functions are identical due to the stationarity condition. A stationary q-th order Markov process can then be fully determined by only one $(q+1)$-

dimensional copula function and a univariate marginal distribution. Estimating these two functions is much easier. However, while the stationarity condition simplifies the problem, it limits the application of the CHOMP model. Non-stationary time series cannot be modeled by this stationary CHOMP model.

5.2.2 Modified CHOMP for the Non-Stationary Case

Theorem 5.5 and Corollary 5.6 show that a higher order Markov process can be fully determined by a set of copula functions and a set of univariate marginal distributions. In particular, when the process is stationary, it is fully determined by just one copula function and one univariate marginal distribution. However, both Theorem 5.5 and Corollary 5.6 have limitations from a practical perspective. If one wants to estimate a CHOMP, using Theorem 5.5, there are too many multivariate copula functions to estimate, which makes the estimation almost infeasible. If using Corollary 5.6, the added stationarity condition simplifies the model estimation, but is sometimes too strong to be met in reality. In summary, Theorem 5.5 makes the CHOMP theoretically applicable to any kind of time series, but practically infeasible to estimate. Corollary 5.6, on the other hand, makes the CHOMP very easy to estimate, but limits its application only to stationary time series.

Ideally, we want something in the middle of Theorem 5.5 and Corollary 5.6. In other words, we want a CHOMP model that is relatively easy to estimate, but has weaker conditions that could allow for a broader range of applications. For that purpose, we give Corollary 5.7 which is one of our main contributions in Chapter 5. The new CHOMP model based on Corollary 5.7 is easier to estimate compared to the one in Theorem 5.5, and also has a broader application compared to the one in Corollary 5.6.

Corollary 5.7. *Let $\{X_t\}_{t=1}^{\infty}$ be a real-valued stochastic process, then its c.d.f. sequence, $\{U_i = F_i(X_i)\}_{i=1}^{\infty}$, is a stationary Markov process of order q, $q \geq 1$, if and only if for all $n \geq q+1$,*

$$C_{1,...,n}(u_1, ..., u_n) = \underbrace{C \star^q C \star^q ... \star^q C}_{n-q+1}(u_1, ..., u_n) \tag{5.2.1}$$

where C is a $(q+1)$-dimensional copula such that $C_{i_1,...,i_l} = C_{i_1+h,...,i_l+h}$, $0 \leq h \leq q+1-i_l$, $1 \leq i_1 \leq ... \leq i_l \leq q+1$, $l = 2, ..., q$, and $C_{j_1,...,j_l}$, $1 \leq j_1 \leq ... \leq j_l \leq q+1$, denotes the corresponding marginal of C : $C_{j_1,...,j_l} = C|_{u_i=1, i \neq j_1,...,j_l}$.

The basic idea in Corollary 5.7 follows the splitting idea of copula theory. Using the copula-based characterization, one can split a higher order Markov process $\{X_t\}_{t=1}^{\infty}$ into two parts: the univariate marginals, $\{F_i(X_i)\}_{i=1}^{\infty}$, and the c.d.f. sequence, $\{U_i = F_i(X_i)\}_{i=1}^{\infty}$. If we do not assume stationarity at all, then the resulting CHOMP will be very difficult to estimate practically, as in Theorem 5.5. If we assume stationarity for both the marginals and the c.d.f. sequence, then the stationarity condition will be too strong, as in Corollary 5.6. Therefore it is reasonable to assume stationarity in only one of the two parts. In Corollary 5.7, we impose stationarity only on the c.d.f. sequence. Then the resulting modified CHOMP can be fully determined by only one copula function and a set of univariate marginals. If we want to estimate this modified CHOMP, we only need to estimate one copula function and a set of univariate marginal distributions, which is not very difficult. On the other hand, since we don't impose any assumption on the marginals, the series can have arbitrary profiles, which greatly broadens the application of the modified CHOMP. We give the formal proof of Corollary 5.7 below.

Proof. First of all, the copula function for any set of random variables $X_1, ..., X_n$ is actually the joint c.d.f. for $U_1 = F_1(X_1), ..., U_n = F_n(X_n)$ in $[0,1]^n$, according to page 24 in Nelsen

(2006). Therefore if we treat $\{U_i = F_i(X_i)\}_{i=1}^{\infty}$ as a new process and apply Corollary 5.4, then Equation 5.2.1 holds if and only if $\{U_i = F_i(X_i)\}_{i=1}^{\infty}$ is a stationary Markov process of order q. Then if we can show that $\{U_i = F_i(X_i)\}_{i=1}^{\infty}$ being a Markov process of order q is equivalent to $\{X_t\}_{t=1}^{\infty}$ being a Markov process of order q, we have Corollary 5.7 proved.

Now let us show $\{X_t\}_{t=1}^{\infty}$ is a Markov process of order q, if and only if $\{U_i = F_i(X_i)\}_{i=1}^{\infty}$ is a Markov process of order q.

If $\{X_t\}_{t=1}^{\infty}$ is a Markov process of order q, for any $x_1, ..., x_{n+1} \in R$, one has

$$P(X_{n+1} \leq x_{n+1} | X_n = x_n, ..., X_{n-q+1} = x_{n-q+1}, ..., X_1 = x_1) = \\ P(X_{n+1} \leq x_{n+1} | X_n = x_n, ..., X_{n-q+1} = x_{n-q+1})$$

One also has

$$\begin{aligned} LHS =& \frac{\partial^n F(x_1, ..., x_{n+1})}{\partial x_1 ... \partial x_n} \Big/ \frac{\partial^n F(x_1, ..., x_n)}{\partial x_1 ... \partial x_n} = \frac{\partial^n C(u_1, ..., u_{n+1})}{\partial u_1 ... \partial u_n} \Big/ \frac{\partial^n C(u_1, ..., u_n)}{\partial u_1 ... \partial u_n} \\ =& P(U_{n+1} \leq u_{n+1} | U_n = u_n, ... U_1 = u_1) \end{aligned}$$

and

$$\begin{aligned} RHS =& \frac{\partial^q F(x_{n-q+1}, ..., x_{n+1})}{\partial x_{n-q+1} ... \partial x_n} \Big/ \frac{\partial^q F(x_{n-q+1}, ..., x_n)}{\partial x_{n-q+1} ... \partial x_n} \\ =& \frac{\partial^q C(u_{n-q+1}, ..., u_{n+1})}{\partial u_{n-q+1} ... \partial u_n} \Big/ \frac{\partial^q C(u_{n-q+1}, ..., u_n)}{\partial u_{n-q+1} ... \partial u_n} \\ =& P(U_{n+1} \leq u_{n+1} | U_n = u_n, ..., U_{n-q+1} = u_{n-q+1}) \end{aligned} \tag{5.2.2}$$

Then LHS=RHS indicates $\{U_i = F_i(X_i)\}_{i=1}^{\infty}$ is a Markov process of order q. To show the converse, one just needs to reverse the chain of equalities. Thus Corollary 5.7 is proved. □

Therefore with Corollary 5.7, the modified CHOMP can be fully determined by a copula function and a set of univariate marginals. On one hand, estimating the modified CHOMP is much easier than the original CHOMP in Theorem 5.5, which makes Corollary 5.7 practically more useful. On the other hand, Corollary 5.7 only assumes stationarity in the c.d.f. sequence, and the original process can still have arbitrary marginals. That means only the intrinsic temporal dependence of the series needs to be homogeneous over time, but it can have arbitrary profile. This condition is considerably weaker than what is in Corollary 5.6 where the whole series has to be strictly stationary. In summary, our Corollary 5.7 has weaker condition than Corollary 5.6, but has fewer and easier functions to estimate in practice than Theorem 5.5.

However, compared to Corollary 5.6, the trade-off of using the modified CHOMP in Corollary 5.7 is that now we have multiple marginal distributions to estimate, since the process is no longer stationary. It is not difficult to estimate these univariate marginal distributions, but one will either need multiple observations at each time point, or some additional assumptions will need to be imposed on the sequence. We will give more details on the estimation of this modified CHOMP in Section 5.3.

Now we give the transition c.d.f. and likelihood calculation formula for the CHOMP without assuming any stationarity condition. Let, as before, $\{X_t\}_{t\in T}$ be a Markov process of order q, then for any $t_1 < ... < t_{q+1} \in T$, $x_{t_1}, ..., x_{t_{q+1}} \in R$, and all Borel sets $A \in \mathcal{B}(R)$ of the form $(-\infty, x_{t_{q+1}})$, Ibragimov (2009) gives the transition c.d.f. of the CHOMP as

$$\begin{aligned} &P(X_{t_{q+1}} \in A | X_{t_1} = x_{t_1}, ...X_{t_{q+1}} = x_{t_{q+1}}) \\ &= \frac{\partial^q C_{t_1,...,t_q,t_{q+1}}(F_{t_1}(x_{t_1}), ..., F_{t_q}(x_{t_q}), F_{t_{q+1}}(x_{t_{q+1}}))}{\partial u_{t_1} ... \partial u_{t_q}} \Big/ \frac{\partial^q C_{t_1,...,t_q}(F_{t_1}(x_{t_1}), ..., F_{t_q}(x_{t_q}))}{\partial u_{t_1} ... \partial u_{t_q}}, \end{aligned} \quad (5.2.3)$$

where the numerator on the right hand side of Equation 5.2.3 is the q-th order partial

derivative of the $(q+1)$-dimensional copula for $\{u_{t_1} = F_{t_1}(x_{t_1}), ..., u_{t_{q+1}} = F_{t_{q+1}}(x_{t_{q+1}})\}$, and the denominator is the full derivative of the q-dimensional copula for $\{u_{t_1} = F_{t_1}(x_{t_1}), ..., u_{t_q} = F_{t_q}(x_{t_q})\}$.

Similar to Equation 5.1.4, given a sequence of values $(x_1, ..., x_n) \in R$, the likelihood based on the copula characterization can also be derived by using the Chapman-Kolmogorov equation. Let $f(x_1, ..., x_n)$ denote the likelihood that the data,$(x_1, ..., x_n)$, are generated by the CHOMP with order q whose transition c.d.f.s are given in Equation 5.2.3, then the likelihood can be written as

$$\begin{aligned} f(x_1, ..., x_n) =& \frac{c_{n-q,...,n}(u_{n-q}, ..., u_n)}{c_{n-q,...,n-1}(u_{n-q}, ..., u_{n-1})} \cdot ... \cdot \frac{c_{1,...,q+1}(u_1, ..., u_{q+1})}{c_{1,...,q}(u_1, ..., u_q)} \\ & \cdot c_{1,...,q}(u_1, ..., u_q) \cdot \prod_{i=1}^{n} f(x_i) \end{aligned} \tag{5.2.4}$$

where $c_{\omega-q,...,\omega}(u_{\omega-q}, ..., u_\omega) = \partial C_{\omega-q,...,\omega}(F_{\omega-q}(x_{\omega-q}), ..., F_\omega(x_\omega))/\partial u_{\omega-q}...\partial u_\omega$, $\omega = q+1,...,n$, and $f(x_i)$ is the corresponding one-dimensional marginal density.

Now for the modified CHOMP(q) in Corollary 5.7, the transition c.d.f. and likelihood calculation can be given similarly except for the fact that we now have a common $(q+1)$-dimensional copula function due to the stationarity in the c.d.f. sequence. Using the notation in Equation 5.2.3, we have the following transition c.d.f. for the CHOMP(q):

$$\begin{aligned} & P(X_{t_{q+1}} \in A | X_{t_1} = x_{t_1}, ...X_{t_{q+1}} = x_{t_{q+1}}) \\ & \quad = \frac{\partial^q C^{CM}_{1,...,q,q+1}(F_{t_1}(x_{t_1}), ..., F_{t_q}(x_{t_q}), F_{t_{q+1}}(x_{t_{q+1}}))}{\partial u_{t_1}...\partial u_{t_q}} / c^{CM}_{1,...,q}(F_{t_1}(x_{t_1}), ..., F_{t_q}(x_{t_q})) \end{aligned} \tag{5.2.5}$$

where $C^{CM}_{1,...,q,q+1}$ is the common $(q+1)$-dimensional copula function that controls the dependence structure in the CHOMP(q), $c^{CM}_{1,...,q}$ is the copula density function for the first q dimensions of $C^{CM}_{1,...,q,q+1}$.

Using the same notation as in Equation 5.2.4, the likelihood of the modified CHOMP(q) can be written as

$$f(x_1, ..., x_n) = \frac{c^{CM}_{1,...,q+1}(u_{n-q}, ..., u_n)}{c^{CM}_{1,...,q}(u_{n-q}, ..., u_{n-1})} \cdot ... \cdot \frac{c^{CM}_{1,...,q+1}(u_1, ..., u_{q+1})}{c^{CM}_{1,...,q}(u_1, ..., u_q)} \cdot c^{CM}_{1,...,q}(u_1, ..., u_q) \cdot \prod_{i=1}^{n} f(x_i) \tag{5.2.6}$$

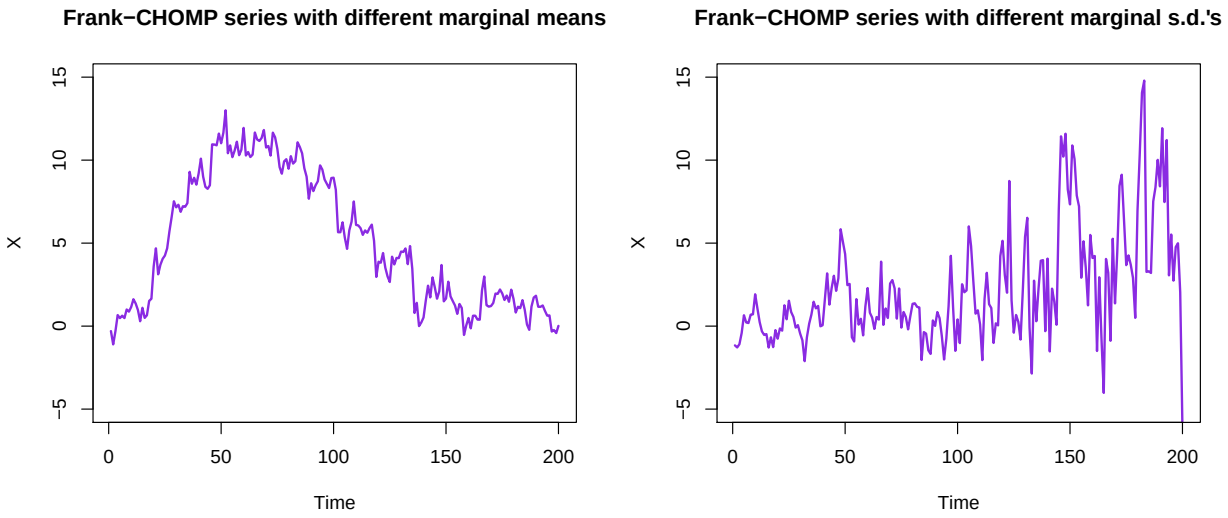
where $c^{CM}_{1,...,q+1}$ is the $(q+1)$-dimensional common copula density function that controls the temporal dependence of the CHOMP(q), $c^{CM}_{1,...,q}$ is the copula density function of the first q dimensions in $c^{CM}_{1,...,q+1}$, and $f(x_i)$ is the corresponding marginal density.

5.2.3 Practical Behavior of Some CHOMPs

Now let us look at the practical behavior of the time series generated by CHOMP. From now on, unless otherwise specified, all the CHOMPs discussed are modified so that they satisfy the conditions in Corollary 5.7. That is, they are all Markov processes with stationary c.d.f. sequences.

As mentioned in Section 5.2.2, one advantage of the modified CHOMP over the original CHOMP by Ibragimov (2009) is that it can generate data with arbitrary shapes by controlling the marginal means and s.d.s at each time point. We give two examples in Figure 5.1. They are generated by two different Frank-CHOMP(1) models. The Frank-CHOMP(1) uses the parametric Frank copula to control the temporal dependence and has first order dependence. Since here we only consider the marginal behavior, the two generating CHOMP models are set to have the same copula function, but different marginals distributions. In Figure 5.1a, the generating CHOMP uses Gaussian marginals with different means at each time point, and a constant s.d. of 1. The marginal means are generated from a scaled Gamma(4,1) density, so

the generated series in Figure 5.1a has a gamma-like shape. The second generating CHOMP, displayed in Figure 5.1b, also has Gaussian marginals with different means, but these means are generated from a scaled Gamma(10,1) density. That is why we see a different shape in Figure 5.1b. In fact, since we can use any values for the means, the series generated by our CHOMP model can have any kind of shape. The second generating CHOMP also uses different s.d.s at each time point and that is why, in Figure 5.1b, we see much more variable behavior later in the series. Moreover, the CHOMP model can use distributions other than the Gaussian distribution, and it can even use different distributions at different time points. This flexibility makes the modified CHOMP suitable for a much larger group of time series compared to the original CHOMP proposed by Ibragimov (2009). Another important feature



(a) (b)

Figure 5.1: CHOMP series with arbitrary shapes

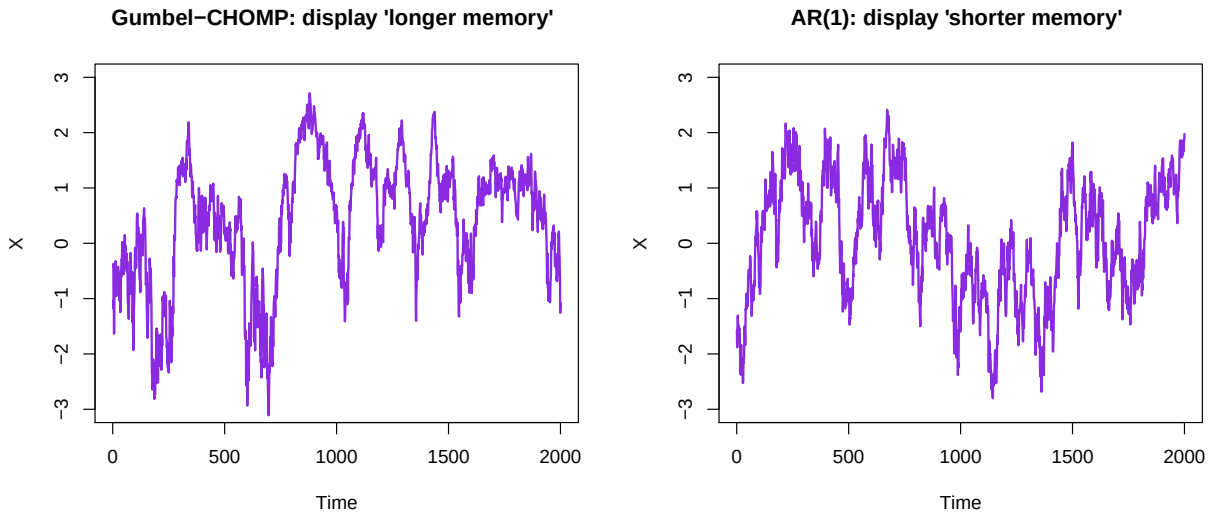
The marginal means in Figure 5.1a are generated by $50 \times Gamma(4,1)$, and the s.d. is 1. The means in Figure 5.1b are generated by $50 \times Gamma(10,1)$, and the s.d.s increase from 1 to 5.

of the CHOMP is that it can exhibit non-linear dependence structures. This is also the reason why we prefer CHOMP over other models that can only handle linear temporal dependence, such as the autoregressive (AR) model. In a CHOMP, the dependence structure is controlled

by the copula function used. In theory, the copula function can account for almost any type of dependence structure, some of which are discussed in Section 4.2 and Section 4.3. However, it is still not very clear how the dependence information in copula functions is reflected in the actual behavior of the time series generated by the CHOMP. In order to see this, we use the Gumbel copula as an example. The Gumbel copula exhibits upper tail dependence as discussed in Section 4.2. We generate a CHOMP(1) series using the Gumbel copula and plot it in Figure 5.2. For comparison, we also generate a time series from AR(1). Since we only want to focus on the dependence structure, we would want dependence strength of the two series to be the same. To do this, we use the Gumbel copula and Gaussian copula with dependence strengths equivalent to a Kendall's τ of 0.875. That gives the Gumbel copula with parameter 8 for the Gumbel-CHOMP(1) and a linear correlation of 0.981 for the AR(1). The second can be done because AR(1) is essentially a CHOMP(1) using Gaussian copula. In addition, we also want the marginal information of the two generating models to be the same. To do that, we use $N(0,1)$ for all marginal distributions in the Gumbel-CHOMP(1). For the AR(1), we adjust the error s.d. to be $\sqrt{1-0.981^2}$ so that the stationary distribution of AR(1) will also be $N(0,1)$. This is because, in AR(1), the variance of the stationary distribution is a function of the error s.d. and the AR coefficient in the following form

$$var(X_t) = \frac{(\text{error s.d.})^2}{1-(\text{AR coefficient})^2}$$

From Figure 5.2, we can see that the time series generated by the Gumbel-CHOMP(1) seems to have a long-range dependence, although, in theory, it is just a first-order Markov process. This observation is due to the upper tail dependence exhibited by the Gumbel copula because the upper tail dependence essentially says that the correlation becomes stronger in extreme values. On the other hand, in Figure 5.2b, although we can also see some dependence exhibited by AR(1), it has a much shorter memory. In this comparison, the two different



(a) (b)

Figure 5.2: Different dependence structures as reflected in actual time series behaviors

The two generating models have the same temporal dependence strength and marginal information. The only difference is the dependence structure. In Figure 5.2a, the temporal dependence is controlled by a Gumbel copula which exhibits a non-linear upper tail dependence, whereas in Figure 5.2b the temporal dependence is linear.

behaviors are caused precisely by the different dependence structures, since we have already controlled the dependence strengths and the marginal information to be the same. Therefore we see that the dependence structure is an important factor in controlling the time series behavior. This behaviorial difference is even more clear from the auto-correlation plot in Figure 5.3. In Figure 5.3, we can see that the auto-correlation of AR(1) has a much faster decay rate. This can be easily understood because the dependence in AR(1) is linear, so it decays at an exponential rate. By contrast, the decay rate for the Gumbel-CHOMP(1) is much slower.

These plots further show that the CHOMP is able to account for not only the profile and the dependence strength of a time series, but also its dependence structure, no matter it is

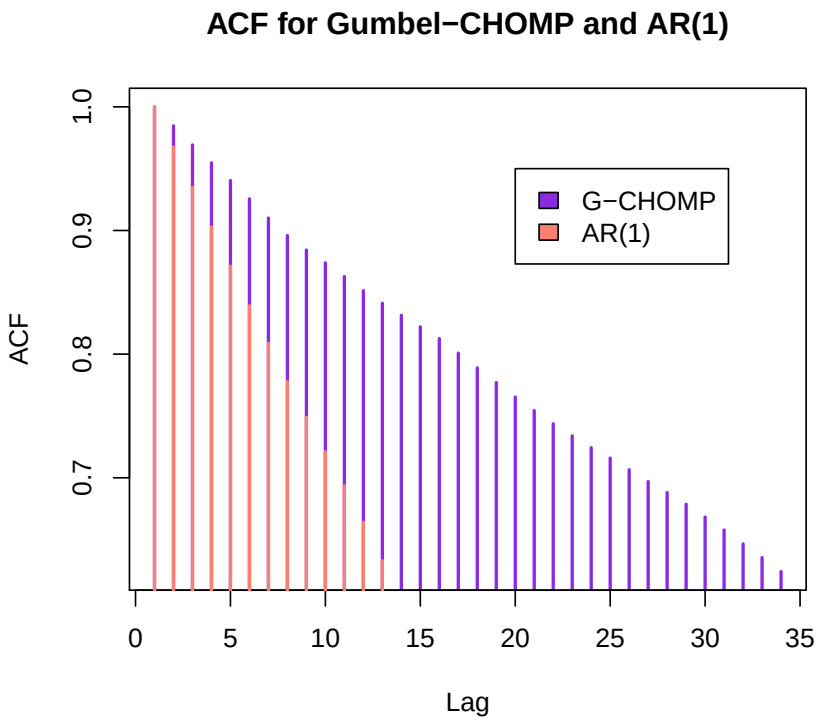


Figure 5.3: Dependence structure could significantly influence the auto-correlation decay rate

linear or non-linear. This allows us to model a larger realm of time series data. Moreover, using CHOMP in the model-based time series clustering, we are also able to extract more useful information from the data, which could greatly benefit the clustering procedure as discussed in Chapter 2.

5.3 Statistical Inferences for CHOMPs

In this Section, we will discuss statistical inference for the CHOMP, including model estimation, likelihood calculation, and the choice of copula. From now on, all the processes

we discuss are assumed to satisfy all the conditions in Corollary 5.7. That is, they are all Markov processes, and their c.d.f. sequences are stationary.

5.3.1 The Two-Step Estimation Procedure

According to Corollary 5.7, a CHOMP model is fully determined by two parts, a copula function and the marginal distribution at each time point. Therefore estimating a CHOMP is equivalent to estimating a copula function and a set of marginal c.d.f.s. Chen and Fan (2006) proposes a two-step estimation procedure to estimate the marginal part and the copula part sequentially in a semi-parametric first order Markov chain. We use the same idea here. Specifically, suppose a time series, X_1, ..., X_T, follows a CHOMP(q) satisfying the conditions in Corollary 5.7. The two estimation steps are: (1) estimate all the univariate marginal distributions, $F_1(X_1)$, ..., $F_T(X_T)$; (2) calculate the estimated c.d.f. sequence, $\{\hat{U}_t = \hat{F}_t(X_t)\}_{t=1}^{T}$, which will later be used to estimate the (q+1)-dimensional copula function.

Marginal Estimation

Let us first consider the stationary case. Suppose a time series, X_1, ..., X_T, follows a stationary CHOMP(q) (the one considered in Corollary 5.6). Since the series is stationary, all the marginal distributions are identical and every data points in the series can be used for estimation. The empirical c.d.f. is one common way to non-parametrically estimate a univariate distribution. We employ a slightly adjusted, but asymptotically equivalent version of the empirical c.d.f. suggested by Chen and Fan (2006). Let $\hat{F}^{AE}$ be the adjusted empirical c.d.f. which can be written as

$$\hat{F}^{AE}(x) = \frac{1}{n+1}\sum_{i=1}^{n} I\{X_i <= x\} \tag{5.3.1}$$

The empirical c.d.f. is scaled by $\frac{1}{n+1}$ and thus avoids having exactly 0's and 1's which often cause numerical difficulties in the computation. Omelka et al. (2009) also claims that using the adjusted empirical c.d.f. provides a better estimation for the copula function in their Monte Carlo simulations.

Now let us consider the non-stationary situation in which the time series follow a CHOMP(q) whose c.d.f. sequence is stationary (as described in Corollary 5.7). In this case, we have a problem because, to estimate a distribution, one will need multiple observations from the same distribution. However, this is usually not feasible for a single time series that is non-stationary as the observations at each time point could be from different distributions. There are potentially two solutions to this problem: One can use certain methods to make the time series stationary. Popular methods include taking a first or second order difference, or fitting a model like ARIMA to make the residuals stationary. Alternatively, one needs to have multiple time series generated from the same model, or a time series that is periodic.

Neither solution is perfect. The drawbacks of the first is that there are a lot of subjective decisions involved in determining which method to use, what model to fit, etc. Also, the first approach implicitly assumes that the marginal distributions at each time point are of the same type. For example, the distributions at different time points can have different means and variances, but they all have to be the same type of distribution. One cannot have normal distribution at one time point and t distribution at another time point. This incurs another assumption that is hard to check.

The drawback of the second solution is also obvious, that is, one doesn't always have multiple time series or periodic time series. However, in certain areas like fMRI and signal processing,

the data series actually can be periodic and sometimes it may not be very hard for researchers to have more data from the same source. Especially in this big data era, getting more data is much easier than in the past and, plausibly, will be even more easier in the future. Moreover, in model-based time series clustering, it is common to assume that there are multiple time series generated from the same model. Otherwise, if every time series is generated from a different model, there is no point in clustering. One may argue that we do not know which time series belong to the same cluster. That is correct, but as mentioned in Section 3.3, the clustering algorithm we develop is an iterative process and will keep relocating different series to different clusters to achieve so that we find the time series that are most likely to be from the same cluster. We give more details about the clustering algorithm in Chapter 6.

Therefore we estimate the marginal distributions assuming there are multiple time series from the same model. Now we have multiple observations from the same distribution at each time point. Then the adjusted empirical c.d.f. in Equation 5.3.1 can be similarly applied. Specifically, let X_{i1}, ..., X_{iT} be the i-th sequence from a CHOMP(q), $1 \leq i \leq n$. Then the marginal distribution at t-th time point can be estimated as

$$\hat{F}_t^{AE}(x) = \frac{1}{n+1} \sum_{i=1}^{n} I\{X_{it} <= x\},\ 1 \leq t \leq T \tag{5.3.2}$$

Copula Estimation

After one has the estimated marginals, the estimated c.d.f. sequence, $\{\hat{U}_t = \hat{F}_t(X_t)\}_{t=1}^{T}$, can be obtained. Given that the original series, X_1, ..., X_n, follows a CHOMP(q) whose c.d.f. sequence is stationary, we can treat the estimated c.d.f. sequence, $\{\hat{U}_i\}_{t=1}^{T}$, as a stationary CHOMP(q). Then every $(q+1)$ consecutive observations in the c.d.f. sequence follow the same copula function. To estimate the copula function, one can use all these

$(q+1)$ consecutive observations. Take the CHOMP(2) for example, if the c.d.f. sequence is of length T, we will have $(T-3)$ three dimensional observations that we can use to estimate the copula function. We give these observations in the following matrix, denoted by $\mathscr{U}$.

$$\mathscr{U} = \begin{pmatrix} \hat{U}_1 & \cdots & \hat{U}_{q+1} \\ \hat{U}_2 & \cdots & \hat{U}_{q+2} \\ \vdots & \ddots & \vdots \\ \hat{U}_{T-q} & \cdots & \hat{U}_T \end{pmatrix} \tag{5.3.3}$$

Each row $\mathscr{U}$ is an observation from the $(q+1)$-dimensional copula function that determines the dependence structure of a CHOMP(q). Therefore the copula function, $C(u_1, ..., u_{q+1})$ can be estimated using the data in Equation 5.3.3.

When $q = 1$, the copula function, $C(u_1, ..., u_{q+1})$, becomes a bivariate copula function. Various parametric and non-parametric bivariate copula estimators have been given in Section 4.2 and Section 4.3. Considering parametric approaches, one can use the regular MLE if the parametric family is already known. Parametric copulas are used by Chen and Fan (2006) in their two-step procedure, which leads to a semi-parametric CHOMP model. If the parametric form is not known, a likelihood-based selection criterion could be used when choosing among a group of parametric copulas. In other words, the parametric copula family that gives the largest likelihood value will be chosen. However, which parametric copulas to try is another subjective decision that needs prior knowledge. Also, the computational cost will be expensive if one tries too many parametric copulas.

Therefore we propose using non-parametric copula estimators in the two-step estimation procedure. Among all the non-parametric copula estimators discussed in Section 4.3, the hybrid beta kernel estimator (Equation 4.3.9) and the improved probit-transformation local likelihood estimator (Equation 4.3.7) have better performance based on our simulation results

in Section 7.3 and the results from Geenens et al. (2014). Let a CHOMP(q) characterized by a set of marginal c.d.f.s, $\mathbf{F} = (F_1, ..., F_T)$, and a $(q+1)$-dimensional copula, $C(\mathbf{u}) = C(u_1, ..., u_{q+1})$., we summarize the non-parametric estimation for the CHOMP(q) as follows

$$CHOMP(q) \sim \{\mathbf{F}, C(\mathbf{u})\}$$
$$\{\hat{\mathbf{F}}, \hat{C}(\mathbf{u})\} = \{(\hat{F}_1^{AE}, ..., \hat{F}_T^{AE}), \hat{C}^{NP}(\mathbf{u})\}$$

where $\hat{F}_i^{AE}$ is the adjusted empirical c.d.f. (Equation 5.3.1) for the i-th time point, and $\hat{C}^{NP}(\mathbf{u})$ stands for the suggested non-parametric copula estimator, including $\hat{C}^{hbk}$ in Equation 4.3.9 and $\hat{(C)}^{ptloc(p)}$ in Equation 4.3.7.

When $q \geq 2$, due to the notorious curse of dimensionality, we limit our discussion to the Archimedean copula family. The estimation of the multivariate Archimedean copulas is the same as the bivariate parametric copulas. Its limitation has been discussed in Section 4.2.2.

Now we give the formula for the likelihood calculation of a CHOMP(q) whose estimated copula and marginals are $\{\hat{\mathbf{F}}, \hat{C}(\mathbf{u})\}$, where $\hat{\mathbf{F}} = (\hat{F}_1, ..., \hat{F}_T)$ and $\hat{C}(\mathbf{u}) = \hat{C}(u_1, ..., u_{q+1})$. Given a time series, x_1,..., x_T, the estimated likelihood, $\hat{f}(x_1, ..., x_T)$, that it is generated from the specified CHOMP(q) is given as

$$\hat{f}(x_1, ..., x_T) = \frac{\hat{c}(u_{T-q}, ..., u_T)}{\hat{c}(u_{T-q}, ..., u_{T-1})} \cdot ... \cdot \frac{\hat{c}(u_1, ..., u_{q+1})}{\hat{c}(u_1, ..., u_q)} \cdot \hat{c}(u_1, ..., u_q) \cdot \prod_{t=1}^{q} \hat{f}_i(x_t)$$

where

$$\hat{c}(u_1, ..., u_{q+1}) = \partial \hat{C}(u_1, ..., u_{q+1}) / \partial u_1 ... \partial u_{q+1}$$
$$\hat{c}(u_1, ..., u_q) = \hat{c}(u_1, ..., u_{q+1})|_{u_{q+1}=1}$$
$$\hat{f}_t(x) = \partial \hat{F}_t(x) / \partial x, \ 1 \leq t \leq T$$

Chapter 6

Model-Based Time Series Clustering Using CHOMP

In this Chapter, we describe our proposed model-based time series clustering algorithm using the CHOMP. The actual clustering algorithm may differ, depending on whether or not the series are stationary. For clustering stationary time series, since the CHOMP can be fit to every single series, we employ the distance-based clustering approach mentioned in Section 3.2. For clustering non-stationary time series, the CHOMP can not be fit to a single time series as mentioned in Section 5.3. Therefore we employ the finite mixture model approach and the classification expectation-maximization (CEM) algorithm.

6.1 Clustering Algorithm for Stationary Time Series

Here we assume that all time series to be clustered are generated from stationary CHOMPs with a common order q. We assume q is known, but these processes could have different

temporal dependence controlled by the copulas in their generating CHOMPs. When the time series is stationary, results in Corollary 5.6 directly apply. Then each time series is fully determined by a $(q+1)$-dimensional copula function and a univariate marginal distribution. The two-step estimation procedure for CHOMP can be applied to every single time series as discussed in Section 5.3. From each time series, we have an estimated copula function and an estimated marginal distribution, $\{\hat{C}(\mathbf{u}), \hat{\mathbf{F}}\}$, which contains the complete probabilistic information of the time series. Then we cluster these estimated functions through agglomerative hierarchical clustering. We use the L1 distance and the Kolmogorov-Smirnov (KS) distance between the estimated functions and they can be written as

$$D^{L-1}(f_1(x), f_2(x)) = \int_S |f_1(x) - f_2(x)| dx \tag{6.1.1}$$

$$D^{KS}(f_1(x), f_2(x)) = \sup_S |f_1(x) - f_2(x)| \tag{6.1.2}$$

where S is the common support of $f_1(x)$ and $f_2(x)$.

Now we propose our stationary time series clustering algorithm using CHOMP in the following way. The proposed algorithm combines the distance-based algorithm mentioned in Section 3.2 with the CHOMP model. Suppose one has a data set consisting of n time series of length T (they do not have to be the same length, but for convenience, we assume they are) and they all follow CHOMP(q), the clustering algorithm proceeds as follows

1. Choose the number of clusters, k.

2. Fit the CHOMP to each time series using the two-step procedure.

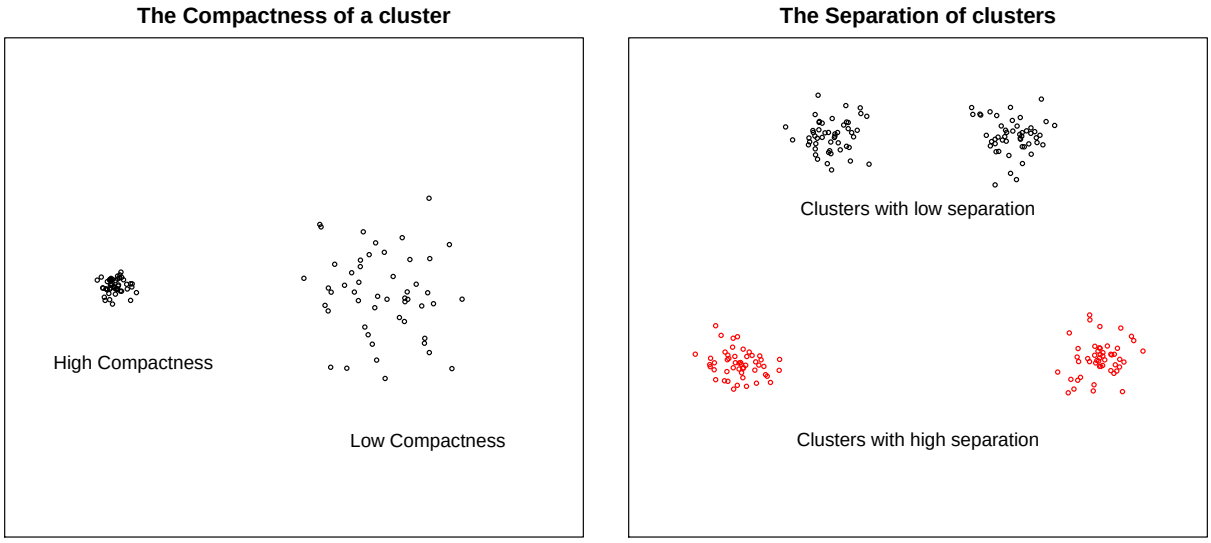
3. Obtain the model outputs which are the estimated copula functions: $\{\hat{C}_i(\mathbf{u}), \hat{\mathbf{F}}_i\}$, $i = 1, ..., n$. They can be rewritten as $\hat{C}_i(u_1 = \hat{F}_1(x_1), ..., u_n = \hat{F}_T(x_T))$.

4. Calculate the distance matrix among all the estimated copula functions using the KS

distance or the L1 distance.

5. Apply agglomerative hierarchical clustering to the distance matrix and cut the clustering tree at k clusters.

In order to choose the number of clusters, a number of internal clustering validity measures can be used. We use the two popular validity measures: the silhouette width and the Dunn index. These validity measures combines the compactness and separation of the cluster partitions. Compactness evaluates the homogeneity within clusters, while separation assesses how far apart clusters are from each other. Figure 6.1 gives a visual explanation of the two validity measures. An ideal clustering will have high compactness within clusters, and high



(a) (b)

Figure 6.1: Compactness and separation of the cluster partitions

separation between clusters. Both the silhouette width and the Dunn index are developed based on this idea.

The silhouette width is the average of each object's silhouette value. For object i, its silhouette width is defined as

$$sil(i) = \frac{b_i - a_i}{\max(b_i, a_i)}$$

where a_i is the average distance between object i and all other observations in the same cluster, and b_i is the average distance between i and the observation in the nearest neighboring cluster. Specifically,

$$a_i = \frac{1}{card(\mathscr{C}(i))} \sum_{j \in \mathscr{C}(i)} dist(i,j)$$
$$b_i = \min_{\mathscr{C}^\star \setminus \mathscr{C}(i)} \sum_{j \in \mathscr{C}_r} \frac{dist(i,j)}{card(\mathscr{C}_k)}$$

where $\mathscr{C}(i)$ is the cluster containing the i-th object, $card(\mathscr{C})$ is the number of objects in cluster $\mathscr{C}$, $\mathscr{C}_r$ is the r-th cluster, and $\mathscr{C}^\star$ is the set containing all the clusters. Therefore the silhouette value can roughly be treated as the ratio of compactness to separation. A well-clustered object has a silhouette value near 1, and a poorly clustered object has a silhouette value near -1. Thus the silhouette width also lies in $[-1, 1]$ and a larger silhouette width indicates a better clustering. The silhouette width can be calculated by the *silhouette* function in the statistical software R.

The Dunn index can be computed as

$$Dunn(\mathscr{C}^\star) = \frac{\min_{\mathscr{C}_r, \mathscr{C}_s \in \mathscr{C}^\star, \mathscr{C}_r \neq \mathscr{C}_s} \left(\min_{i \in \mathscr{C}_r, j \in \mathscr{C}_s} dist(i,j) \right)}{\max_{\mathscr{C}_w \in \mathscr{C}^\star diam(\mathscr{C}_w)}} \tag{6.1.3}$$

where $diam(\mathscr{C}_w)$ is the maximum distance between objects in cluster $\mathscr{C}_w$. The Dunn index is the ratio of the smallest distance between objects not in the same cluster to the maximum within-cluster distance. The Dunn index has a value between zero and $+\infty$, and a larger value indicates a better clustering. But the Dunn index is less robust than the silhouette

width to extreme situations. For example, if there are two very close clusters, one will have a very small Dunn index even if all other clusters are very well separated. This is because the Dunn index is trying to make the two closest clusters separate enough, which might sometimes be too conservative. See Brock et al. (2008) and the references therein for more details about the validity measures.

To choose the number of clusters, we run the proposed clustering algorithm with different values of k, and then choose the one giving the largest silhouette width or Dunn index.

6.2 Clustering Algorithm for Non-Stationary Time Series

Now we consider clustering non-stationary time series. We assume these time series are generated from CHOMPs that satisfy the conditions in Corollary 5.7. In other words, these time series could have arbitrary shapes, but their c.d.f. sequences are generated from stationary CHOMPs with a common order q. But these time series can have different dependence strengths or structures, and we assume q to be known.

According to Corollary 5.7, each time series can be fully determined by a $(q+1)$-dimensional copula function and a set of marginal distributions. However, as mentioned in Section 5.3, since each time series is non-stationary, the marginal distributions at different time points are different. Therefore within a single time series, there is only one observation for each marginal distribution. This makes it impossible to estimate all the marginal distributions using one single time series, and the CHOMP model cannot be estimated using just one single time series. Thus we cannot apply the distance-based clustering algorithm like we do in Section 6.1.

One solution provided in Section 5.3 is to use multiple time series from the same generating model when estimating the CHOMP. This solution is feasible when we employ the finite mixture model approach to cluster the time series. This is because, in the finite mixture model setting, all the time series are assumed to be generated by a relatively small number of models. Therefore there will always be multiple time series from the same generating model.

However, there still is a problem. That is, we do not know which time series are from the same model at the beginning. Before we explain how our algorithm solves this problem, we would first like to give our proposed model-based time series clustering algorithm using CHOMP. The proposed algorithm combines the CHOMP model with the finite mixture model, and uses an adaptation of the classification expectation-maximization (CEM) algorithm by Celeux and Govaert (1992) to find the best clustering.

First, let us give the settings of the finite mixture model in our case. Let $x_i(t)$ denote the observation at the t-th time point of i-th time series, $i = 1, ..., n$, $t = 1, ..., T$. We assume they are generated by M different CHOMPs that correspond to the M clusters of interest. Let $z_{ik} = \{1, 0\}$ be the cluster indicator for the i-th time series to the k-th cluster, e.g. $z_{1,3} = 1$ means object 1 belongs to cluster 3. Let $l_k(\mathbf{x}_i|C_k(\mathbf{u}), \mathbf{F}_k)$ be the conditional likelihood that data object $\mathbf{x}_i$ is generated from the k-th CHOMP. Let p_k be the prior probability that a time series is from the k-th CHOMP such that, each $\mathbf{z}_i$ is i.i.d. from a multinomial distribution of one draw from M categories with probabilities, $p_1, ..., p_M$. The conditional log-likelihood of the entire mixture model, $L((\mathbf{x}_1, ..., \mathbf{x}_n))$, can then be expressed as

$$L((\mathbf{x}_1, ..., \mathbf{x}_n)|\Xi) = \sum_{i=1}^{n} \sum_{k=1}^{M} z_{ik} \log[l_k(\mathbf{x}_i|C_k(\mathbf{u}), \mathbf{F}_k)]$$

where $\Xi = \{C_1(\mathbf{u}), ..., C_M(\mathbf{u}), \mathbf{F}_1, ..., \mathbf{F}_M, z_{11}, ..., z_{nM}\}$ and $\sum_{k=1}^{M} p_k = 1$. Our purpose is to find Ξ and z_{ik} such that the conditional log-likelihood of the entire mixture model is maximized. We then adapt the CEM algorithm proposed by Celeux and Govaert (1992) to the above setting.

The algorithm starts with a given number of clusters M, and an initial clustering $\mathscr{C}^\star$. The γ-th iteration of CEM ($\gamma > 0$) is defined as follows:

E-step: compute for $i = 1, ..., n$ and $k = 1, ..., M$ the current posterior probabilities $ptr_k^m(\mathbf{x}_i)$ that $\mathbf{x}_i$ belongs to the k-th CHOMP:

$$ptr_k^m(\mathbf{x}_i) = \frac{p_k^m l_k(\mathbf{x}_i|\hat{C}_k^m(\mathbf{u}), \hat{\mathbf{F}}_k^m)}{\sum_{k'=1}^{M} p_{k'}^m l_{k'}(\mathbf{x}_i|\hat{C}_k^m(\mathbf{u}), \hat{\mathbf{F}}_k^m)}$$

where p_k^m is the mixture probability for the k-th CHOMP and $\{\hat{C}_k^m(\mathbf{u}), \hat{\mathbf{F}}_k^m\}$ stands for the estimation of the k-th CHOMP at the current iteration m.

C-step: assign each $\mathbf{x}_i$ to the cluster which provides the maximum posterior probability. If the maximum posterior probability is not unique, we choose the cluster with the smallest index. Let $\mathscr{C}^{\star m}$ be the new resulting clustering.

M-step: for $k = 1, ..., M$, obtain $\{\hat{C}_k^{m+1}(\mathbf{u}), \hat{\mathbf{F}}_k^{m+1}\}$ by the two-step procedure (discussed in Section 5.3) using the time series in cluster k. The mixture probabilities are estimated as

$$p_k^{m+1} = \frac{card(\mathscr{C}_k^m)}{n}, \ k = 1, ..., M$$

We use a non-informative prior for the mixture probabilities. Celeux and Govaert (1992) prove the convergence of the CEM algorithm. The difference between CEM and ordinary EM is that the CEM algorithm incorporates a classification step between the E-step and the

M-step using a maximum a posteriori (MAP) principle. This classification step allows the CEM to directly find the clustering result, whereas in the regular EM, one determines the clustering through the probabilities of belonging to each cluster. That is to say, in CEM, each object either belong or not belong to a cluster, but in EM, it belongs to each cluster with a probability.

In addition, as mentioned in Vrac et al. (2012), it is better to omit the mixture probabilities $\{p_k\}$ in the optimization procedure of CEM when the overall focus is more on the clusters than on the modeling of the whole mixture model. Due to this reason, we omit the mixture probabilities $\{p_k\}$, resulting in the following modified CEM:

E-step: compute for $i = 1, ..., n$ and $k = 1, ..., M$ the current posterior probabilities $ptr_k^m(\mathbf{x}_i)$ that $\mathbf{x}_i$ belongs to the k-th CHOMP:

$$ptr_k^m(\mathbf{x}_i) = \frac{l_k(\mathbf{x}_i|\hat{C}_k^m(\mathbf{u}), \hat{\mathbf{F}}_k^m)}{\sum_{k'=1}^{M} l_{k'}(\mathbf{x}_i|\hat{C}_k^m(\mathbf{u}), \hat{\mathbf{F}}_k^m)}$$

where $\{\hat{C}_k^m(\mathbf{u}), \hat{\mathbf{F}}_k^m\}$ stands for the estimation of the k-th CHOMP at the current iteration m.

C-step: assign each $\mathbf{x}_i$ to the cluster which provides the maximum posterior probability. If the maximum posterior probability is not unique, we choose the cluster with the smallest index. Let $\mathscr{C}^{\star m}$ be the new resulting clustering.

M-step: for $k = 1, ..., M$, estimate $\{\hat{C}_k^{m+1}(\mathbf{u}), \hat{\mathbf{F}}_k^{m+1}\}$ by the two-step procedure (discussed in Section 5.3) using the time series in cluster k.

Practically, there will be another problem when executing this algorithm. Although we start with M clusters, sometimes after the C-step, the number of clusters will be less than M.

In other words, one cluster is relocated entirely to another cluster. To mitigate for this situation, we add another iteration in our algorithm as follows:

1. Given the number of clusters M, randomly create an initial clustering assignment.
2. Perform the (modified) CEM algorithm until convergence.
3. If the number of clusters in the final clustering is equal to M or the number of iterations exceeds the pre-specified maximum (100), then algorithm ends. Otherwise go to step 1.

Now we explain how our algorithm can solve the problem raised in the beginning of this Section as well as in the end of Section 5.3. The problem is: we need multiple time series from the same generating model to estimate a non-stationary CHOMP. Although multiple time series from the same model exist in the time series clustering context, we do not know which ones are from the same model, at least not at the beginning. Based on the iterative process in the CEM algorithm, the answer is apparent. It is true that, in the initial clustering, time series in the same cluster do not necessarily belong to the same generating model. However, the CEM algorithm does not end here. It keeps relocating the time series to different clusters in order to achieve the maximum likelihood. Therefore eventually the time series in the same cluster are most likely to belong to the same generating model.

Similar to clustering stationary time series, we determine the number of clusters by optimizing certain criteria. However, instead of the internal validity measures, we can now use likelihood-based criterion because the overall likelihood of the entire mixture model can be estimated. Therefore choosing the right k is very similar to a model selection procedure. According to Fraley and Raftery (2002), we are selecting among mixture models that have different components and/or different number of components.

We repeat the proposed time series clustering algorithm with different numbers of clusters k, and then determine the best k via standard model selection criteria. We employ three commonly used criteria: Akaikes information criterion (AIC) of Akaike (1973), the Bayesian information criterion (BIC) of Schwarz (1978), and the consistent AIC (CAIC) of Bozdogan (1987),

$$AIC = -2 \times loglikelihood + 2p$$

$$BIC = -2 \times loglikelihood + p \log n$$

$$CAIC = -2 \times loglikelihood + p(\log n + 1)$$

where p is the number of parameters, n is the number of time series.

In summary, we propose two model-based time series clustering algorithms using CHOMP, one for stationary time series, and the other for non-stationary time series. The algorithm for stationary time series combines the CHOMP model with agglomerative hierarchical clustering, and the algorithm for non-stationary time series combines the CHOMP model with the finite mixture model and CEM. Although agglomerative hierarchical clustering, the finite mixture model and the CEM are not new, combining them with the CHOMP model provides a new clustering approach that is able to use more information from time series for clustering. Due to the advantages of the CHOMP, the proposed algorithm can cluster time series based on not only their dependence strength, but also their linear or non-linear dependence structure. This versatility makes our algorithm a very attractive alternative for clustering time series. In Chapter 7 and 8, we will show its performance in an extensive simulation study and two real data analyses.

Chapter 7

Simulations

In this Chapter, an extensive simulation study is carried out to evaluate the performance of our proposed model-based time series clustering algorithm using CHOMP. We evaluate the performance of our algorithm using time series data with different properties. The data simulated in this Chapter include: (1) stationary time series with first order linear Markovian dependence; (2) stationary time series with first order non-linear Markovian dependence; (3) non-stationary time series with second order linear Markovian dependence; (4) non-stationary time series with second order non-linear Markovian dependence. Table 7.1 gives the generating models of these time series.

Table 7.1: Time Series Simulation Models

Name	Dependence Order	Stationary	Dependence Pattern
AR(1)	1	Yes	Linear
Frank-CHOMP(1)	1	Yes	Non-linear
Frank-CHOMP(2)	2	No	Non-linear
Gaussian process	2	No	Linear

AR(1)

The first order autoregressive model (AR(1)) in Table 7.1 is a commonly used time series model. The time series generated by AR(1) are stationary Markov chains with first-order linear dependence. Let X_t be the random variable at time point t of the time series, then one can formulate the AR(1) model as follows:

$$X_{t+1} = \alpha + \beta X_t + \epsilon_{t+1}$$

where α is the intercept, $\epsilon_t \sim N(0, \sigma^2)$ for all t. The stationary distribution for the AR(1) is $N(\frac{\alpha}{1-\beta}, \frac{\sigma^2}{1-\beta^2})$. We simulate the AR(1) time series by using the *arima.sim* function in R.

Frank-CHOMP

In Table 7.1, both the stationary Frank-CHOMP(1) and non-stationary Frank-CHOMP(2) are special cases of the CHOMP model. Their non-linear dependence structures are controlled by the bivariate Frank copula and the three dimensional Frank copula, respectively. We simulate the time series from these two models by using the transition c.d.f. given in Equation 5.2.5. For the stationary Frank-CHOMP(1), we use the bivariate Frank copula and the standard normal c.d.f. in place of the copula function and marginal c.d.f.s in Equation 5.2.5. For the non-stationary Frank-CHOMP(2), we use a three-dimensional Frank copula and normal c.d.f.s with different means in place of the copula function and marginal c.d.f.s in Equation 5.2.5. As a result the stationary Frank-CHOMP(1) generates stationary time series with first order non-linear Markovian dependence, and the non-stationary Frank-CHOMP(2) generates non-stationary time series with second order non-linear Markovian dependence. We will give more details about these two generating models in Section 7.1 and Section 7.2. The generating Frank-CHOMPs used are purely parametric models since they use both a parametric copula and the normal c.d.f. for the marginals.

Gaussian process

The Gaussian process in Table 7.1 is a non-stationary second order Markov chain. In this process, we let every three consecutive random variables follow a multivariate Gaussian distribution. Therefore, if a time series of length T is generated by the Gaussian process, there will be $T-2$ three-dimensional Gaussian distributions involved. These multivariate distributions are allowed to have different means and s.d.s, but they are required to have a common correlation matrix. Therefore the time series generated by this Gaussian process will have the second order Markovian property and arbitrary profiles, but its temporal dependence will be linear and homogeneous over time. More details about this model will be given in Section 7.2.

7.1 Clustering Stationary Time Series

In this Section, we evaluate the performance of our proposed algorithm in clustering real-valued stationary time series. Specifically, we would like to see: (1) how well can it cluster series with linear dependence and (2) how well can it cluster series with non-linear dependence.

We achieve our goal by conducting two sets of simulation experiments. In the first set of experiments, we use AR(1) to simulate stationary time series with linear dependence. Then we see how well the proposed algorithm can cluster them, and then compare its performance with an AR(1)-based clustering algorithm. In the second set of experiments, we use the Frank-CHOMP(1) to simulate stationary time series with a special non-linear dependence. Again, we compare the performance of our algorithm with the AR(1)-based clustering algorithm.

7.1.1 Simulating AR(1) Time Series

According to the literature of model-based time series clustering in Section 3.2, real-valued time series are usually clustered using ARIMA and dynamic regression models. For stationary time series, one of the most representative models is the AR(p). All of these models only allow linear temporal dependence. One interesting fact is that the AR(p) is actually a special case of both the ARIMA model and the dynamic regression model according to Frühwirth-Schnatter (2011). This is because both the ARIMA model and the dynamic regression model use the autoregressive part to model the temporal dependence. Then in terms of modeling the temporal dependence, these models are essentially the same as the AR(p). Therefore, for simplicity, we will just use the AR(1) model to simulate stationary time series with linear dependence. Then the simulated data will be used for clustering. Now we give more details about the simulation settings and how we simulate the AR(1) time series.

In this Section, we conduct a total of three simulations. All the AR(1) time series in these simulations are simulated using the *arima.sim* function in R. Also, their stationary distributions are all set to be $N(0, 1)$, and they have the same length of 500.

In the first simulation, we generate two groups of time series from the AR(1) model. Each group has 50 time series. But the two groups are generated with different AR coefficients: 0.1 and 0.15. In the second simulation, we also generate 50 time series for each group, but the two AR coefficients are changed to 0.3 and 0.4. In other words, the temporal dependence of these series is stronger and the difference between the two groups is also larger. In the third simulation, each group still has 50 series, and the two AR coefficients are 0.85 and 0.9. The temporal dependence is even stronger. We summarize these simulation settings in Table 7.2.

Table 7.2: Simulation Settings for AR(1) Models

Sim. No.	Runs	Group	Mean/SD	AR coef.	# of TS	Length
1	100	1	0/1	0.1	50	500
		2	0/1	0.15	50	500
2	100	1	0/1	0.3	50	500
		2	0/1	0.4	50	500
3	100	1	0/1	0.85	50	500
		2	0/1	0.9	50	500

7.1.2 Simulating Frank-CHOMP(1) Time Series

In the second set of experiments, we simulate time series with non-linear dependence using the stationary Frank-CHOMP(1). These generated time series also have Markovian property and are stationary. But their temporal dependence is now non-linear, and is controlled by the bivariate Frank copula function given in Table 4.1.

Specifically, we generate these time series by using the following transition c.d.f.:

$$P(X_{t+1} \leq x_{t+1} | X_t = x_t) = \frac{\partial C^F(u_t, u_{t+1}; \theta)}{\partial u_t}$$

where $C^F(\cdot, \cdot; \theta)$ is the bivariate Frank copula function, $u_t = \Phi(x_t)$, and Φ is the standard normal cumulative distribution function. The initial observation can be generated from a $N(0, 1)$. A convenient way to randomly generate a time series from the Frank-CHOMP(1) is to first generate its c.d.f. sequence $\{u_t\}$ using the following conditional probability:

$$P(U_{t+1} \leq u_{t+1} | U_t = u_t) = \frac{\partial C^F(u_t, u_{t+1}; \theta)}{\partial u_t}$$

where $C^F(\cdot, \cdot; \theta)$ is the bivariate Frank copula with parameter θ. Then we use the inverse standard normal c.d.f., Φ^{-1}, to obtain the original sequence $\{x_t\}$. Note that, since we use

the bivariate Frank copula and standard normal marginals, the Frank-CHOMP(1) is now a parametric model with one parameter (the Frank copula parameter, θ).

In this Section, we will conduct two simulations to generate time series from the Frank-CHOMP(1). The generated time series have the same length of 500, and their marginal distributions are all $N(0,1)$, which means their stationary distributions are also $N(0,1)$.

In the first simulation, we generate two groups of time series from the Frank-CHOMP(1). Each group has 50 time series. The two groups of time series are generated with different Frank parameters (θ): 1 and 1.5. In the second simulation, we also generate two groups of time series. Each group has 50 time series, and the two Frank parameters are 20 and 25. Therefore in the second simulation, the overall temporal dependence is much stronger. We summarize the simulations settings in Table 7.3.

Table 7.3: Simulation Settings for Frank-CHOMP(1) Model

Sim. No.	Runs	Group	Mean/SD	Frank Param.	# of TS	Length
1	100	1	0/1	1	500	50
		2	0/1	1.5	500	50
1	100	1	0/1	20	500	50
		2	0/1	25	500	50

7.1.3 Simulation Results for Clustering Stationary Time Series

Since the data generated in these simulations are all stationary, we use the algorithm proposed in Section 6.1, which combines the CHOMP model with the distance-based clustering algorithm. In this case, we use a semi-parametric Frank-CHOMP(1) in the clustering algorithm, which assumes a Frank parametric form for the copula and non-parametric form for the marginals. Thus the Frank-CHOMP(1) used in the clustering algorithm is semi-parametric and is different from the Frank-CHOMP(1) used to simulate the time series.

Specifically, we fit a semi-parametric Frank-CHOMP(1) model to each time series and then cluster the estimated Frank parameters by agglomerative hierarchical clustering. For comparison, we apply the same distance-based clustering algorithm with the AR(1) model. That is, we fit an AR(1) model to each time series and then cluster the estimated AR coefficients through agglomerative hierarchical clustering. For now we assume the number of clusters and the order of dependence are known.

Therefore the two competing approaches use the same clustering algorithm. But they use different models to extract information from the time series. On the other hand, since the time series data are simulated by the Frank-CHOMP(1) and AR(1), we would expect that when clustering time series generated by AR(1) models, the algorithm using AR(1) should have better performance, and when clustering time series generated by Frank-CHOMP(1)s, our algorithm should perform better.

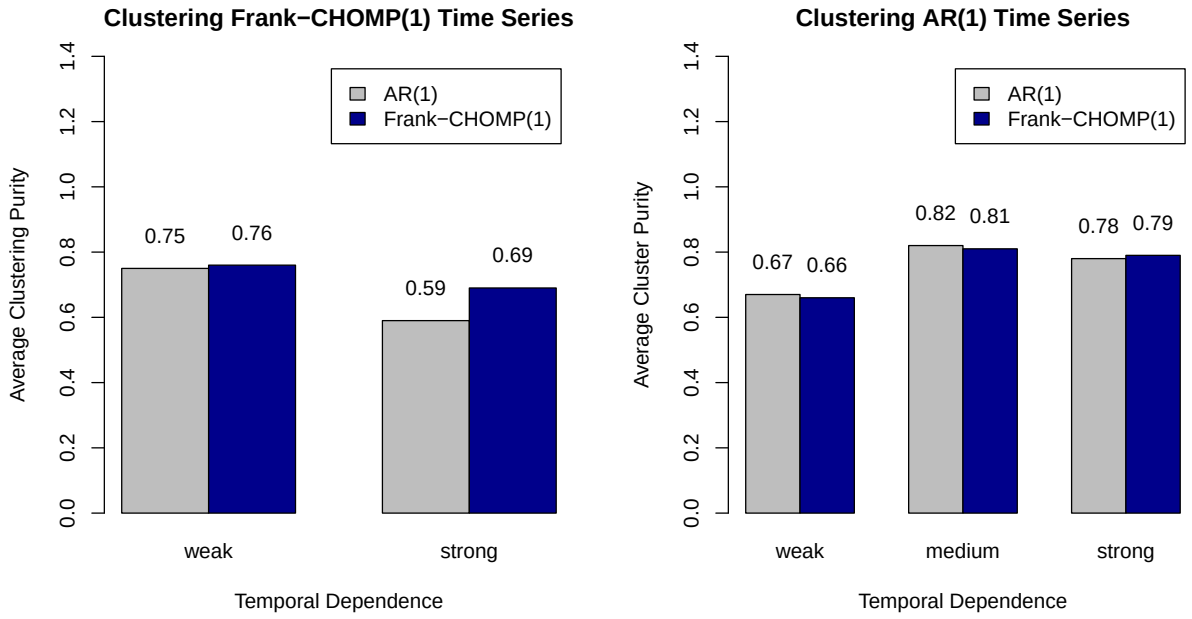
However, in the clustering results, we find that our algorithm performs equally well to the AR(1)-based algorithm when clustering series generated by AR(1), and outperforms it when clustering series generated by the Frank-CHOMP(1).

We evaluate the clustering results by calculating the clustering purity. For each clustering result, the clustering purity can be calculated as

$$\frac{\text{number of correctly clustered time series}}{\text{total number of time series}} \tag{7.1.1}$$

Each algorithm is applied to each data set for 100 simulation runs. Then we calculate the average clustering purity for each case. These results are summarized in Figure 7.1.

Figure 7.1 shows that our algorithm works better than the AR(1)-based algorithm in clustering sequences that exhibit Frank copula dependence, especially when the dependence is strong. This is as expected because the Frank-CHOMP(1) used in our algorithm is designed



(a) (b)

Figure 7.1: Average clustering purity of 100 simulations with stationary sequences

for this type of dependence, hence it is the "home game" for our algorithm. What really makes our algorithm stand out is Figure 7.1b where the sequences exhibit AR dependence. This is now the "home game" for the AR(1) model, but our algorithm works equally well as the AR(1)-based algorithm, regardless of the dependence strength.

In summary, from these simulations, our algorithm using the Frank-CHOMP(1) is shown to be able to cluster time series with both linear and certain non-linear dependence, while the algorithm using AR(1) is only suitable for clustering time series with linear dependence. In fact, as mentioned earlier, most models used in the literature of model-based time series clustering only allow linear temporal dependence. Therefore when certain types of non-linear dependence exist, our proposed clustering algorithm using semi-parametric CHOMP gives

better results than using AR(1) because it is able to capture more accurate information from the temporal dependence of the series.

One may argue that we need to know beforehand what type of non-linear dependence it is to determine which parametric copula to use. For example, the Frank-CHOMP(1) used here can only handle non-linear dependence exhibited by the Frank copula. If the data exhibit another type of non-linear dependence, we will need to use another copula function. This problem is overcome by employing the non-parametric CHOMP in our proposed clustering algorithm. The corresponding simulation studies are given in Section 7.3. However, even if we use the semi-parametric CHOMP, it is still better than using models like ARIMA and dynamic regression models in terms of modeling the temporal dependence. This is because if we use the ARIMA or dynamic regression model, we would not have the flexibility to model any non-linear temporal dependence even if we know what type of non-linear dependence it is.

7.2 Clustering Non-Stationary Time Series

In Section 7.1, the simulations focus on clustering stationary time series with first order Markovian dependence. Now we evaluate the performance of our algorithm when clustering non-stationary time series with higher order Markovian dependence. Since the time series are non-stationary, we will use the algorithm proposed in Section 6.2. The algorithm combines the CHOMP model with the finite mixture model and uses an adaptation of the CEM to find the optimal clustering. Since here our focus is not the estimation of the overall mixture model, we will use the modified CEM by omitting the mixture probabilities.

Unlike the stationary time series, non-stationary time series have more information in the data: the temporal dependence information and the sequence profile information. The sequence profile information is induced because the marginal distributions at each time point are no longer identical. Therefore our goals here are to see: (1) can our algorithm utilize the sequence profile information in time series to benefit the clustering? (2) can our algorithm utilize the higher order linear dependence information in time series to benefit the clustering? (3) can our algorithm utilize the higher order non-linear dependence information to benefit the clustering?

To answer these questions, we conduct two sets of simulation experiments. In the first set of experiments, we simulate groups of time series from the non-stationary Guassian process with second order Markovian property. In the second set of experiments, we simulate groups of time series from the Frank-CHOMP(2) model that has a second order non-linear dependence controlled by a three-dimensional Frank copula. In each simulation, we compare the performance of our algorithm with k-means clustering that can only capture the profile information of the time series. The AR(p) model is not used here because it only applies to stationary time series.

7.2.1 Simulating Time Series from the Gaussian Process

In this Section, we use the Gaussian process to generate non-stationary time series with second order linear Markovian dependence. To do that, we let every three consecutive random variables follow a three dimensional multivariate Gaussian distribution with different means and s.d.s. These different means and s.d.s make the resulting time series non-stationary. However, we want to keep the intrinsic dependence homogeneous over time, otherwise the time series will have dependence change points which is out of our discussion. Therefore we

use a common correlation matrix for all these multivariate Gaussian distributions. Mathematically, suppose X_t is the random variable at time point t following $N(\mu_t, \sigma_t)$, then we let, for all t,

$$(X_t, X_{t+1}, X_{t+2}) \sim N((\mu_t, \mu_{t+1}, \mu_{t+2}), \mathbf{\Sigma}_t)$$

where

$$\mathbf{\Sigma}_t = \begin{pmatrix} \sigma_t^2 & \sigma_t\sigma_{t+1}\rho & \sigma_t\sigma_{t+2}\rho \\ \sigma_t\sigma_{t+1}\rho & \sigma_{t+1}^2 & \sigma_{t+1}\sigma_{t+2}\rho \\ \sigma_t\sigma_{t+2}\rho & \sigma_{t+1}\sigma_{t+2}\rho & \sigma_{t+2}^2 \end{pmatrix}, \ \mathbf{Corr}_t = \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix}$$

With some matrix operations, one has

$$X_{t+2}|(X_t = x_t, X_{t+1} = x_{t+1}) \sim N(\mu_{t+2|(t,t+1)}, \sigma^2_{t+2|(t,t+1)}) \tag{7.2.1}$$

where

$$\mu_{t+2|(t,t+1)} = \mu_{t+2} + \mathbf{\Sigma}_{t,21}\mathbf{\Sigma}_{t,22}^{-1} \begin{pmatrix} x_t - \mu_t \\ x_{t+1} - \mu_{t+1} \end{pmatrix}$$

$$\Sigma_t = \begin{pmatrix} \Sigma_{t,22} & \Sigma_{t,21} \\ \Sigma_{t,12} & \Sigma_{t,11} \end{pmatrix}$$

$$\sigma^2_{t+2|(t,t+1)} = \Sigma_{t,11} - \Sigma_{t,21}\Sigma_{t,11}^{-1}\Sigma_{t,12}$$

Therefore, given μ_t, σ_t and ρ, we are able to simulate a time series using the conditional normal c.d.f. in Equation 7.2.1. We use the same ρ here due to the limitation of multivariate Archimedean copulas discussed in Section 4.2.2.

We conduct two simulations. In the first simulation, we generate three groups of time series with a common length of 100. Each group has a common sequence of means, but different sequences of s.d.s. Let μ_t^i and σ_t^i be the mean and s.d. for the i-th group at the t-th time

point, and ρ^i be the correlation in the correlation matrix for the i-th group, $i = 1, 2, 3$, $t = 1, ..., 100$. We then set μ_t^i, σ_t^i and ρ^i as follows in the first simulation:

$$\mu_t^i = 0.02 \times t, \text{ for } i = 1, 2, 3.$$
$$\sigma_t^i \sim |N(i, 0.5)|, \text{ for } i = 1, 2, 3.$$
$$\rho_1 = 0.1, \rho_2 = 0.5, \rho_3 = 0.9$$

This makes the time series from the different groups have a common profile (controlled by $\{\mu_t^i\}$), but different temporal dependence (controlled by $\{\rho_i\}$).

In the second simulation, we use similar settings. But now we make time series from different groups have different profiles and different temporal dependence. Specifically, we set μ_t^i, σ_t^i and ρ_i as follows in the second simulation

$$\mu_t^i = \frac{(i+1)}{100} \times t, \text{ for } i = 1, 2, 3.$$
$$\sigma_t^i \sim |N(i, 0.5)|, \text{ for } i = 1, 2, 3.$$
$$\rho_1 = 0.1, \rho_2 = 0.5, \rho_3 = 0.9$$

We summarize these settings in Table 7.4.

Table 7.4: Simulation Settings for the Gaussian Process

Sim.	Runs	Group	Means	S.D.s	Corr.
1	100	1	(1,2,...,100)× 0.02	$\|N(1, 0.5)\|$	0.1
		2	(1,2,...,100)× 0.02	$\|N(2, 0.5)\|$	0.5
		3	(1,2,...,100)× 0.02	$\|N(3, 0.5)\|$	0.9
2	100	1	(1,2,...,100)× 0.02	$\|N(1, 0.5)\|$	0.1
		2	(1,2,...,100)× 0.03	$\|N(2, 0.5)\|$	0.5
		3	(1,2,...,100)× 0.04	$\|N(3, 0.5)\|$	0.9

7.2.2 Simulating Frank-CHOMP(2) Time Series

Now we conduct another set of experiments to simulate non-stationary time series with second order non-linear Markovian dependence by using the Frank-CHOMP(2) model. The second order non-linear dependence is exhibited by using the three dimensional Frank copula. Mathematically, the time series can be generated by using the following transition c.d.f.:

$$P(X_{t+1} \leq x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}) = \frac{\partial^2 C^F(u_{t-1}, u_t, u_{t+1})}{\partial u_{t-1} \partial u_t} / c^F_{1,2}(u_{t-1}, u_t) \tag{7.2.2}$$

where $C^F(\cdot,\cdot,\cdot)$ is the three-dimensional Frank copula function, $c^F(\cdot,\cdot)_{1,2}$ is the corresponding Frank copula density for the first two dimensions, $u_t = F_t(x_t)$, and $x_t \sim N(\mu_t, \sigma_t^2)$. Therefore, given μ_t, σ_t, and the Frank copula parameter, θ, we are able to simulate a time series following the conditional c.d.f. in Equation 7.2.2. Since both the copula function and the marginals are parametric, the Frank-CHOMP(2) used to simulate the data is a purely parametric model.

We perform two simulations. In the first simulation, we generate three groups of time series from the Frank-CHOMP(2). They have a common length of 100. Time series from different groups are set to have different sequence profiles determined by the means at each time point. For simplicity, we use the same s.d. for all time points. The sequence profile is set to be a straight line (one can use different means to set any kind of shape, but the point here is to have different profiles among the groups). The temporal dependence is now determined by the Frank copula parameter. We also let this parameter be different among the groups. Therefore for the first simulation, we have

$$\mu_t^i = \frac{(i+1)}{100} \times t, \ i = 1, 2, 3, t = 1, ..., 100.$$
$$\sigma_t^i = 1, \ i = 1, 2, 3, t = 1, ..., 100.$$
$$\theta_1 = 0.5, \theta_2 = 4.5, \theta_3 = 8.5$$

In the second simulation, we use similar settings to generate three groups of time series from the Frank-CHOMP(2). However, now we set the time series from different groups to have the same sequence profile and only different temporal dependence. The corresponding parameters are given as follows:

$$\mu_t^i = 0.02 \times t,\ \ i = 1, 2, 3, t = 1, ..., 100.$$

$$\sigma_t^i = 1,\ \ i = 1, 2, 3, t = 1, ..., 100.$$

$$\theta_1 = 0.5, \theta_2 = 4.5, \theta_3 = 8.5$$

We summarize the settings for these two simulations in Table 7.5.

Table 7.5: Simulation Settings for Mixture of Frank-CHOMP(2)s

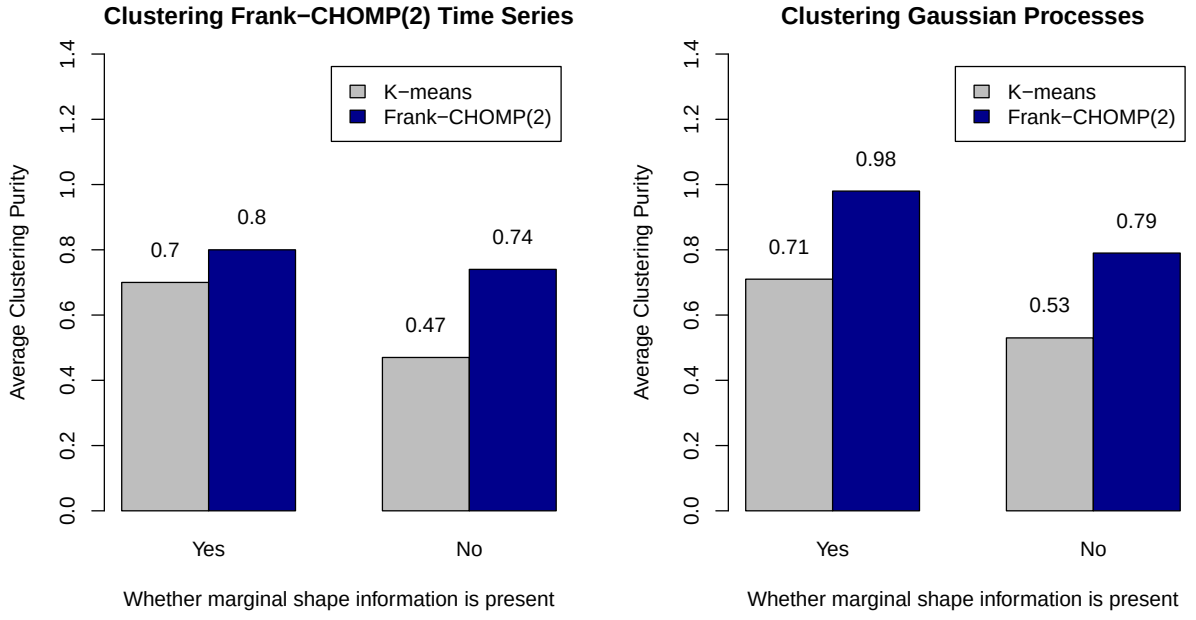
Sim. No.	Runs	Group	Mean/SD	Frank Param.	Length	No. of TS
1	100	1	$(1, ..., 100) \times 0.02/1$	0.5	100	50
		2	$(1, ..., 100) \times 0.03/1$	4.5	100	50
		3	$(1, ..., 100) \times 0.04/1$	8.5	100	50
2	100	1	$(1, ..., 100) \times 0.02/1$	0.5	100	50
		2	$(1, ..., 100) \times 0.02/1$	4.5	100	50
		3	$(1, ..., 100) \times 0.02/1$	8.5	100	50

7.2.3 Simulation Results for Non-stationary Time Series

Now we have simulated two sets of time series data. One data set contains series with higher order linear Markovian dependence generated from the Gaussian process, and the other contains time series with higher order non-linear Markovian dependence from Frank-CHOMP(2)s. In each set of data, we also have one subset with sequence profile information to be used for clustering and the other subset without.

We apply our proposed clustering algorithm using CHOMP. Since the simulated time series now are non-stationary, we apply the clustering algorithm proposed in Section 6.2. The CHOMP model used here is the semi-parametric Frank-CHOMP(2) which assumes a three dimensional Frank parametric form for the copula and adjusted empirical c.d.f. for the marginals and is different from the parametric model used to generate the data.

As discussed in Section 6.2, this algorithm uses both the profile information and the temporal dependence information, hence gives better results than algorithms that only used one of them, such as k-means clustering. Therefore we expect our proposed algorithm to perform better than k-means clustering, especially when time series from different groups have the same sequence profile. We apply both our algorithm and k-means clustering. The clustering results are summarized in Figure 7.2. In Figure 7.2a, we can see that our proposed algorithm



(a) (b)

Figure 7.2: Average clustering purity of 100 simulations with non-stationary sequences

has an average clustering purity of 0.8, while k-means clustering has a purity of 0.7. In this

case, the sequence profiles are different among the true clusters, so profile information in the data can be used for clustering. That is why k-means clustering has a clustering purity of 0.7, which is an acceptable value. But our algorithm has a higher purity of 0.8, so our algorithm must have used other information which benefits the clustering procedure. According to the simulation setting in Table 7.5, that other information can only be the different temporal dependence among the true clusters. Since the data are generated by the Frank-CHOMP(2), the temporal dependence here is second-order non-linear Markovian dependence. Based on these observations, we can answer two of the questions raised at the beginning of Section 7.2. That is, our proposed algorithm using semi-parametric CHOMP can utilize both the profile information and certain non-linear temporal dependence information for clustering. Similar conclusions can be drawn from Figure 7.2b. Therefore our proposed algorithm using semi-parametric CHOMP can also utilize linear temporal dependence information for clustering.

In summary, these simulations show that our proposed algorithm using CHOMP can use both the profile information and the temporal dependence information for clustering. With more useful information available, the clustering results are significantly improved. In addition, our proposed algorithm can also use higher order dependence information for clustering.

One can have arguments similar to those raised in the end of Section 7.1 about the limitations caused by the parametric form of the copula used in the CHOMP model. Our answers will be the same as those in Section 7.1. That is, in terms of modeling the temporal dependence, the semi-parametric CHOMP is at least better than models that can only handle linear dependence.

7.3 Clustering Using Non-Parametric CHOMP

Based on the simulation results in Section 7.1 and Section 7.2, we see that our proposed algorithm using the semi-parametric CHOMP is able to capture multiple types of information from the data to cluster time series, including the profile, linear temporal dependence and certain non-linear temporal dependence. With more information available, our algorithm outperforms its competitors that can only use part of the information in the data.

However, as mentioned at the end of Section 7.1, our algorithm's ability is limited by the fixed form of the copula function in the semi-parametric CHOMP. Before we give our solution, let us make the problem a little more clear. The temporal dependence information in a time series actually consists of two parts: the dependence strength and the dependence structure. For example, in Section 7.1, all the time series clustered in each simulation are generated by the same model type, either AR(1) or Frank-CHOMP(1). Therefore the generated time series might have different dependence strengths, but the actual structure of the temporal dependence is the same, either the AR dependence or the Frank dependence. That means, the differences among those clusters are the dependence strengths, not the structure of the dependence. Our clustering algorithm using the semi-parametric CHOMP is better than the AR(1)-based algorithm because it is able to detect these dependence strengths no matter it is in the linear structure or the non-linear Frank structure. However, if one wants to detect the difference between two types of dependence structures, our algorithm using the semi-parametric CHOMP may not work as well. This is because the semi-parametric CHOMP uses parametric copulas which cannot model multiple dependence structures.

To solve this problem, there are two solutions: we can use a likelihood-based criterion to select the most appropriate parametric copula every time we fit a semi-parametric CHOMP, or we can use a non-parametric CHOMP. The first solution is just to try different copula

functions when fitting a CHOMP. If two time series choose different parametric copulas, then their dependence structures might very well be different. However, this approach is still limited by the number of considered copula functions, and the choice of the copula list can also be subjective. Therefore we instead use the non-parametric copula, which lead to non-parametric CHOMPs.

The difference between the semi-parametric CHOMPs and the non-parametric CHOMPs is just the type of copula used. In non-parametric CHOMPs, we do not assume a specific form for the copula function. Estimating a non-parametric CHOMP thus is equivalent to estimating a non-parametric copula and a set of marginal c.d.f.s. The marginal c.d.f.s will still be estimated using the adjusted empirical c.d.f. and we use multiple methods to non-parametrically estimate the copula function. See Section 5.3.

Now we conduct another set of simulation experiments. Our goal in these experiments is to show that our proposed algorithm using the non-parametric CHOMP is able to distinguish between the structures of the temporal dependence in time series, and then cluster based on that. Since we want to focus on the structure of the temporal dependence, we set all the simulated data to be stationary time series with the stationary distribution $N(0, 1)$ so that they have very similar profile information.

In both simulations, we generate two groups of time series with very similar dependence strengths. However, one group of time series is generated by the AR(1) model, and the other group is generated by the Frank-CHOMP(1) model. Thus the time series between groups have similar dependence strengths, but different dependence structures. Now we explain the difference between the settings of the two simulations.

In the first simulation, the settings are controlled in a way so that it is more difficult for the Frank-CHOMP(1) based algorithm to classify the two groups. We do this through the

following procedure: first, we generate an AR(1) time series with coefficient 0.95 and fit a Frank-CHOMP(1) to it. The estimated Frank parameter is about 15.74 (depending on the length of simulated series, and random number seeds, coefficient might differ slightly). Then we generate another time series from the Frank-CHOMP(1) with the same Frank parameter 15.74. This way, the two resulting series will have similar dependence strength (at least in terms of the Frank parameter), but obviously different dependence structures. When one mixes these two types of time series together and fits a Frank-CHOMP(1) model to them, the estimated Frank coefficients will be almost identical. Therefore in this case, Frank-CHOMP(1) based clustering algorithm should not be effective. On the other hand, if one switches the order of the process and first generates time series with the Frank-CHOMP(1), the resulting series are challenging for the AR(1)-based algorithm. This is exactly the setting in the second simulation. Therefore neither of the models is expected to work well in both situations because neither of them are able to capture the distinctions between the structures of the temporal dependence. But the proposed algorithm using the non-parametric CHOMP should perform well in both situations. We summarize the simulation settings in Table 7.6.

Table 7.6: Simulation Settings for the Mixture of AR(1) and Frank-CHOMP(1)

Sim.	Runs	Which first	Length	Group	Model	Coef.
1	100	AR(1)	500	1	AR(1)	0.95
				2	F-CHOMP(1)	15.74
2	100	F-CHOMP(1)	500	1	AR(1)	0.95
				2	F-CHOMP(1)	29

We apply our proposed algorithm using the non-parametric CHOMP(1) to the simulated data. Since the time series are stationary, we can use the distance based clustering algorithm proposed in Section 6.1. A CHOMP will be estimated in each time series, and the estimated copula densities will then be used for clustering (we estimate the copula density which is equivalent to the copula function). The copula density is estimated by a total of five different

non-parametric methods discussed in Section 4.3.

- the hybrid Beta kernel copula ("hbk")
- the mirror-reflection kernel copula ("Mirror")
- the probit-transformation kernel copula ("pt1")
- the improved probit-transformation copula using linear local likelihood ("ptloc1")
- the improved probit-transformation copula using quadratic local likelihood ("ptloc2")

Then the estimated copula densities are clustered by agglomerative hierarchical clustering. The distance matrix is calculated by using the KS distance in Equation 6.1.2 and the L1 distance in Equation 6.1.1. We assume the number of clusters and the order of dependence are known. The simulation results are evaluated by the clustering purity in Equation 7.1.1. The average clustering purities from 100 simulation runs are presented in Figure 7.3 and Figure 7.4.

In Figure 7.3, the data were simulated such that it is harder for the semi-parametric Frank-CHOMP(1) based algorithm to cluster. Therefore we see in the left two bars that the average clustering purity for the AR(1) based algorithms (0.88) is higher than that for the Frank-CHOMP(1)-based algorithm (0.63). In Figure 7.4, the situation for the simulated data is the opposite, so we see the average clustering purity for the AR(1) based algorithms (0.53) is lower than that for the Frank-CHOMP(1)-based algorithm (0.77). Therefore neither AR(1) nor Frank-CHOMP(1) based algorithm work well in both situations. This is because they only look at the strength of a particular dependence type, hence are not able to capture the real structural differences in the temporal dependence.

Our proposed algorithm using non-parametric CHOMP, on the other hand, is able to capture both the strength and the structure of the temporal dependence.

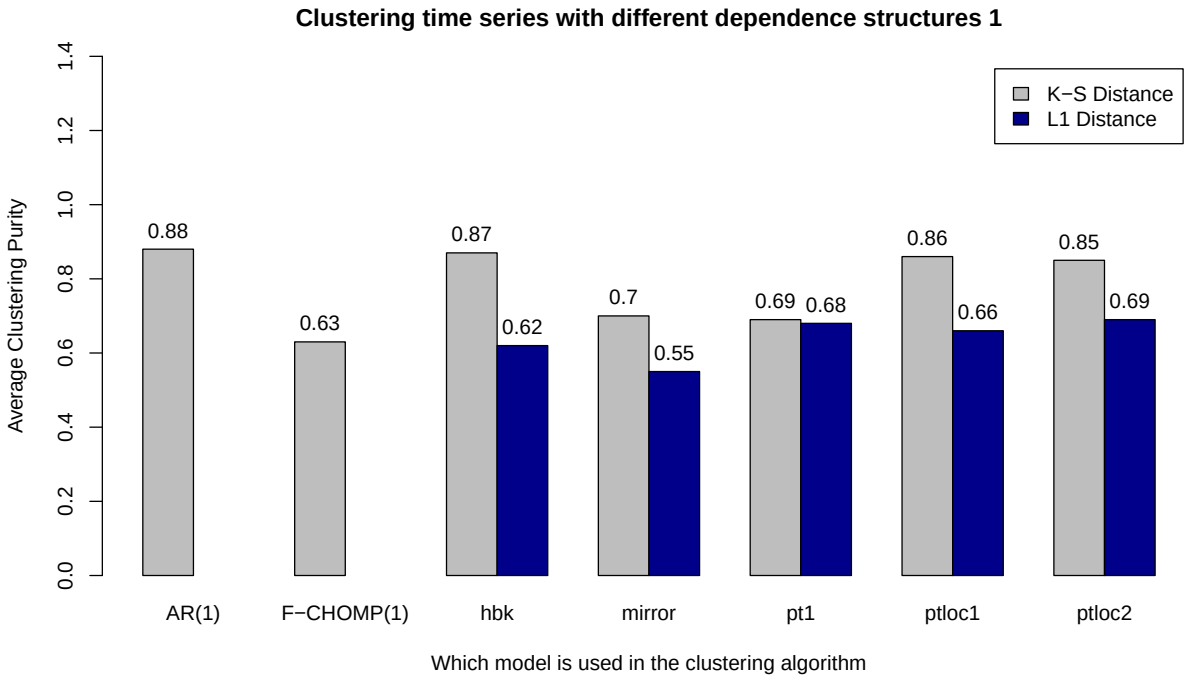


Figure 7.3: First simulation experiment using non-parametric CHOMP(1)s

The simulation is set up to make it challenging for the Frank-CHOMP(1)-based clustering algorithm and easy for the AR(1)-based clustering algorithm. The non-parametric CHOMP(1)-based clustering algorithm works well using KS distance and any of the non-parametric copulas "hbk","ptloc1" and "ptloc2"

Now let us look at the performance of our algorithm using the non-parametric CHOMP(1). Based on the results, we see that if we pick the right non-parametric copula and distance measure, our algorithm using the non-parametric CHOMP(1) will perform well in both situations, while the other two algorithms only work well in one of the two situations. But how can we pick the right copula and distance measure? We explain this in the next few paragraphs.

From Figure 7.3, our non-parametric CHOMP(1) based algorithm works well if we use the KS distance and any of the non-parametric copulas "hbk", "ptloc1" and "ptloc2". Their corresponding average clustering purities are 0.87, 0.86 and 0.85 which are very similar to the

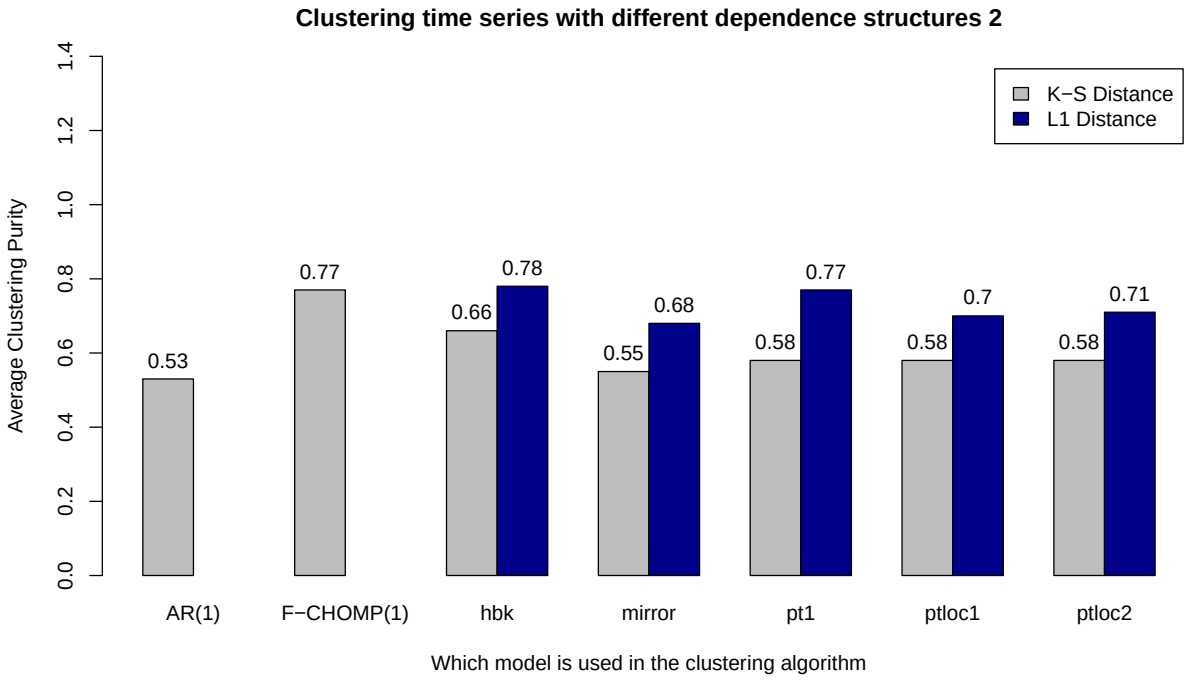
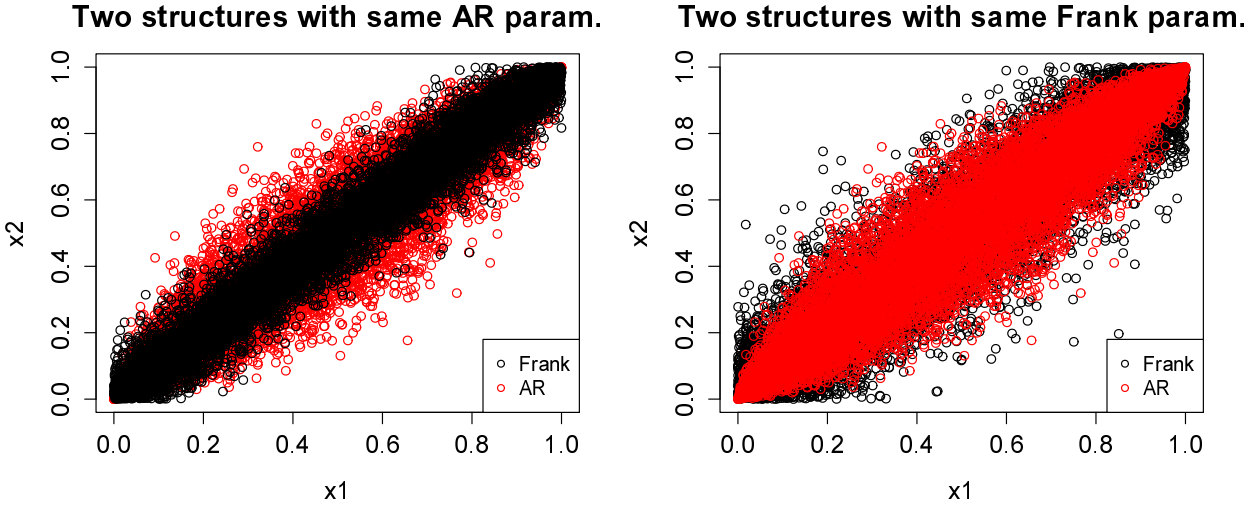


Figure 7.4: Second simulation experiment using non-parametric CHOMP(1)s

The simulation is set up to make it challenging for the AR(1)-based clustering algorithm and easy for the Frank-CHOMP(1)-based clustering algorithm. The non-parametric CHOMP(1)-based clustering algorithm works well using L1 distance and any of the non-parametric copulas "hbk" and "pt1".

one (0.88) for the AR(1)-based algorithm. From Figure 7.4, our non-parametric CHOMP(1) based algorithm works well if we use the L1 distance and any of the non-parametric copulas "hbk" and "pt1". Their corresponding average clustering purities are 0.78 and 0.77 which are very similar to the one (0.77) for the Frank-CHOMP(1)-based algorithm. When using the "ptloc1" and "ptloc2" copula estimator in the second set of simulation, the average clustering purity is not quite as good (0.70 and 0.71), but still acceptable. Therefore it seems that the hybrid beta kernel copula ("hbk") and the two improved probit-transformation copulas work well in both of these simulation settings, but the two improved probit-transformation copulas work a little worse in the second set of simulations.

Another interesting observation is that, in the first set of simulations, our non-parametric CHOMP(1) based algorithm always performs better using the KS distance than the L1 distance, but it is the opposite in Figure 7.4. To explain this phenomenon, we look at the temporal dependence exhibited in the simulated time series. Since the time series in the simulations are generated by AR(1) and Frank-CHOMP(1), their temporal dependence is characterized by a bivariate Gaussian copula and a bivariate Frank copula. Therefore we just need to compare these two copula functions. We compare two sets of copulas in Figure 7.5. The first set of copulas corresponds to the two types of temporal dependence structures exhibited in the first set of simulated time series. The second set of copulas corresponds to the second set of simulations. The dependence strengths of the copulas are set to be equal to those from the first and second sets of simulations. In Figure 7.5a, one can see that the



(a) Dependence structures in the first set of simulations

(b) Dependence Structures in the second set of simulations

Figure 7.5: The difference between two distance measures' detection capabilities

difference between the two copulas is apparent throughout the range, and this is when the L1 distance should work better because it is detecting the average difference of the entire

domain. However, in Figure 7.5b, the most significant difference occurs in the two corners. If one uses the L1 distance for the two types of copulas in Figure 7.5b, the differences at the corners will diminish after averaging over the entire domain.

Now we know why using different distance measures give different clustering performance. Then the question is: how to determine which distance measure to use? In reality we will not be able to calculate the clustering purity because we don't know the true clusters. One possible solution is to use internal validity measures, such as the silhouette width and the Dunn index discussed in Section 6.1. Since the Dunn index is less robust, we use the silhouette width here. Ideally, the distance measure that gives higher clustering purity should have better validity measures. If this is true, we can then choose the distance measure giving better values on these validity measures. To check if this supposition, in Table 7.7, we calculate the silhouette width for the first simulation. We only do this for "hbk", "ptloc1" and "ptloc2" because they have the best performance. From Table 7.7, we can see that

Table 7.7: Average Validity Measures and Clustering Purity from 100 Simulations

Copula	hbk		ptloc1		ptloc2	
Distance	KS	L1	KS	L1	KS	L1
Cluster Purity	0.87	0.62	0.86	0.66	0.85	0.69
Silhouette	0.43	0.14	0.48	0.21	0.44	0.34

the distance measure that gives higher clustering purity also has larger silhouette width. Therefore based on this limited exploration the silhouette width may be useful in helping us choose which distance measure to use.

Table 7.8: Frequency of Detecting the True Number of Clusters in 100 Simulations

Sim. Model	Avg. Purity for 3 clusters	AIC	BIC	CAIC
Gaussian process 1	0.98	6/100	89/100	85/100
Gaussian process 2	0.79	44/100	56/100	46/100

7.4 Determining the Number of Clusters

In all the simulations we have conducted so far, the number of clusters is assumed to be known. Determining the number of clusters is a very difficult topic that has received much attention in the literature (Liao 2007), especially when the true clusters are not very different from each other. Practically, we would also suggest combining quantitative methods with the interpretability of the clusters when choosing the number of clusters. The method we use to determine the optimal number of clusters is given in Section 6.1 for clustering stationary time series and in Section 6.2 for clustering non-stationary time series. Here we only test the method's performance for non-stationary time series.

To determine the optimal number of clusters, we run the algorithm several times with different number of clusters. Model selection criteria including AIC, BIC and CAIC are calculated for choosing the number of clusters. The optimal number of clusters should give the smallest criterion value. We test the effectiveness of these criteria on the data simulated by the non-stationary Gaussian process in Section 7.2 where the true number of clusters is three. We run the clustering algorithm with a range of numbers from one to six. The results in Table 7.8 show how often each criterion detects the true number of clusters in 100 simulation runs. We see that BIC and CAIC are much more effective in detecting the correct number of clusters than AIC. However, when the clustering purity with known number of clusters becomes lower, which means the true clusters are less distinct, the performance in picking

the right number of clusters gets worse for BIC and CAIC. This is as expected because when it is harder to distinguish the time series from each other, it is, of course, also harder to tell how many clusters there are. There is not much we can do if the true clusters are similar to each other. However, sometimes "similar" true clusters occur because of the lack of information. Recall that in the apple and orange example at the first Chapter, if one can only see their shapes, the "true clusters" do look very similar. But when you have more information available, the clustering becomes much easier. This is exactly why our clustering algorithm using the CHOMP is advantageous because it is able to collect more useful information for clustering. In fact, in Chapter 8, we will show in a real data analysis that the number of clusters is much easier to detect compared to an AR(1)-based clustering algorithm.

Chapter 8

Real Data Analysis

In this Chapter, we apply the proposed model-based clustering algorithm using CHOMP to two real data sets. One is the 1929-1999 state personal income data, and the other is resting state fMRI data.

8.1 Personal Income Data

The personal income data set we will use has been studied by Kalpakis et al. (2001) and Xiong and Yeung (2002) for time series clustering. The data consists of time series from 1929-1999 in 25 states of the USA. The values in the time series are the per capita personal income which is the total personal income for divided by the population of the state. Kalpakis et al. (2001) defined all the East Coast states plus CA and IL as group 1, which has a high income growth rate. The remaining states, which are in the mid-west, are defined as Group 2, which has a low income growth rate. Even though it is hard to say that this definition of two groups yields true clusters, it can still be used as a benchmark to compare results. Table 8.1

shows the defined classification of the 25 states by Kalpakis et al. (2001). We perform the

Table 8.1: Classification of the 25 States Defined by Kalpakis et al. (2001)

Group 1	California, Connecticut, Delaware, D.C., Florida, Illinois, Maine, Maryland, Massachusetts, New Jersey, New York, North Carolina, Pennsylvania, Rhode Island, Vermont, Virginia
Group 2	Idaho, Indiana, Iowa, Kansas, Nebraska, North Dakota, Oklahoma, South Dakota, West Virginia

same pre-processing mentioned in Kalpakis et al. (2001). Before clustering, each time series is also normalized because the clustering result given by Kalpakis et al. (2001) is obtained by clustering the normalized data. Then we run our CHOMP-based clustering algorithm on the personal income data set for comparison. Since the time series are not stationary, we need to employ the clustering algorithm proposed in Section 6.2 that combines the CHOMP model with the finite mixture model approach.

To evaluate the performance of our proposed algorithm, we compare the clustering results from multiple clustering algorithms. Kalpakis et al. (2001) use the feature extraction approach to cluster the personal income time series. Among the features they have used, they claim that Linear Predictive Coding(LPC) cepstrum (inverse Fourier transformation of the logarithm of the estimated spectrum of a signal) has the highest discriminative power. Then they use the k-medoids clustering algorithm with Euclidean distance to cluster the time series based on their LPCs, assuming the number of clusters is known. Xiong and Yeung (2002) cluster the same data set using the finite AR mixture model. However, they don't report their final clustering result, so we will not include their results in our comparison. In place of the method by Xiong and Yeung (2002), a third method we include in the comparison is to fit an ARIMA model for each time series and then cluster based on the model coefficients. Kalpakis et al. (2001) claims these time series approximately follow ARMA(1,1). We fit an ARIMA(1,1,1) (ARMA(1,1) cannot be fit for every time series due to non-stationarity) to

each series and then cluster the time series based on the estimated AR and MA coefficients. We use the regular k-means clustering. Therefore the two clustering methods for comparison here are: (1) a feature extraction approach using LPC (2) a model-based clustering algorithm using ARIMA.

The clustering results of the above methods are evaluated by calculating the clustering purity using the benchmark clusters defined by Kalpakis et al. (2001) as true clusters. In Table 8.2, we can see that they all have the same clustering purity. This means the LPC-based feature extraction method and the ARIMA-based methods perform equally well with our proposed algorithm when the time series approximately follow an ARIMA-like model. This is because the time series approximately follow an ARIMA model according to Kalpakis et al. (2001), and the temporal dependence is linear. One might be suspicious about why the results are

Table 8.2: Clustering Purities for the Personal Income Time Series

	Our method	Kalpakis et al. (2001)	ARIMA(1,1,1)
Cluster Purity	0.84	0.84	0.84

exactly the same. In fact, the actual clustering from these methods are slightly different. For example, one method could misclassify object A, and the other could misclassify object B. But they both misclassify only one object, which results in the same clustering purity. Another reason could be that we have just a few objects (only 25 states), which makes it more likely to see identical clustering purities.

In summary, this real data example shows that when the time series have linear temporal dependence, our proposed algorithm perform equally well with other algorithms in the literature. In the next real data example, we will see our algorithm outperforms its competitors when the data contains non-linear temporal dependence.

8.2 Resting State fMRI Data

The second real data we study is the resting state functional magnetic resonance image(fMRI) data collected by Park et al. (2010). In brain imaging, fMRI is a very popular method used to study brain functions. The basic mechanism of this method is as follows. First, When a region of the brain is activated to process certain human activity, the degree of blood oxygenation becomes higher in that region. The change of blood oxygenation level can be reflected by the change of MR signals in the same region due to differences in the magnetic properties of oxygenated and de-oxygenated blood. These MR signals are then detected and processed to create a brain image. For more details about how fMRI data is acquired and processed, see Lindquist (2008) and the references therein. Resting state fMRI data are the fMRI data acquired when the subject is resting and not doing any specific task. Therefore this type of data is usually used to study the baseline or normal status of the brain.

In the resting state experiment performed by Park et al. (2010), the subject is asked to gaze on a fixation point and no stimulus is presented. The MR signals are measured at three axial slices of the brain. An axial slice is the two-dimensional plane when you look down to the brain from above. The most inferior slice is at the frontal sinus (slightly above the eyes). All three slices are 4mm thick and are separated by 10mm from each other. The resolution of each slice is 64×64, so one will have 4096 MR measurements from each slice at each time point. On the other hand, the resting state MR signals are measured every two seconds ($TR = 2$ s) for approximately 50 mins. This results in a total of 1500 two-dimensional images for each slice. According to Park et al. (2010), this time period is much longer than the standard in fMRI data collection, allowing for better estimation of the temporal dependence structure. This is also one of the reasons why we choose this data set because our clustering algorithm is good at capturing the differences in temporal dependence.

Other experimental and pre-processing actions are also taken to improve the quality of the data and remove outliers, linear trends and spatial correlation in the data. See Park et al. (2010) for details. Finally, after pre-processing, the data set consists of 12288 time series (3 slices $\times$ (64$\times$64)) that are approximately stationary. Due to the pre-processing, information other than the temporal dependence are eliminated to the maximum extent. Before we cluster the time series, we also apply a model selection procedure in which we fit AR model with orders from 0 to 20 and the optimal order is picked by examining the BIC values. Those series with optimal order of zero are then excluded because they exhibit very little temporal dependence.

We then apply our non-parametric CHOMP-based clustering algorithm to the pre-processed time series in all three slices. Since the series can be considered stationary, we use the distance-based clustering algorithm proposed in Section 6.2. In other words, we fit a non-parametric CHOMP(1) model to each series and then cluster estimated copula functions. The copula functions are estimated using the hybrid beta kernel method. Agglomerative hierarchical clustering is then applied with KS and L1 distance measures. To determine the number of clusters and which distance measure to use, we perform the algorithm with both distances and multiple number of clusters. Then the decisions are made by choosing the largest silhouette widths. For comparison, we also fit an AR(1) model to each fMRI series and then the estimated AR coefficients are clustered using the same agglomerative hierarchical clustering with the same distances. The clustering results for both methods are summarized in Table 8.3

From Table 8.3, we can see that for our algorithm using the non-parametric CHOMP(1), the average silhouette width with the KS distance is generally higher than the L1 distance. Therefore the KS distance should be a better option for this data. While using the KS distance, results with two clusters give the highest average silhouette width of 0.67. Therefore

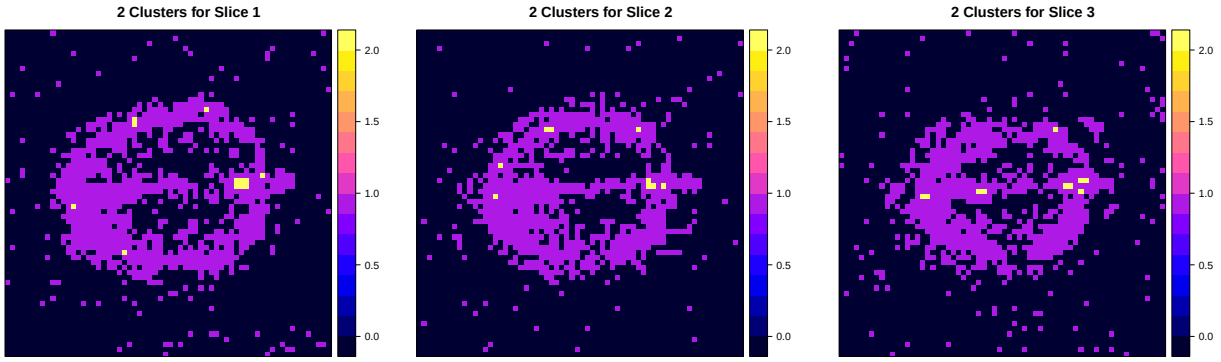
we determine that the number of clusters should be two. The results for the AR(1)-based clustering algorithm also show that the most appropriate number of clusters is two since this is when the average silhouette width is the highest in Table 8.3. However, the average silhouette width for AR(1) (0.63) is slightly lower than CHOMP(1) (0.67). Moreover, using our algorithm, the silhouette width for two clusters is much higher than for other numbers of clusters (0.67 as opposed to 0.48, 0.42, 0.42 and 0.40). But for the AR(1) based algorithm, the distinction is much smaller (0.63 as opposed to 0.58, 0.57, 0.54, 0.53). This observation shows that with our algorithm, detecting the clusters is much easier than using the AR(1)-based clustering algorithm.

Table 8.3: Validity Measures for the Resting State Time Series

Distance	Number of clusters	Avg. silhouette width	
		CHOMP(1)	AR(1)
KS	**2**	**0.67**	0.63
	3	0.48	0.54
	4	0.42	0.53
	5	0.42	0.58
	6	0.40	0.57
L1	2	0.54	same as those for KS
	3	0.37	
	4	0.24	
	5	0.24	
	6	0.17	

We plot the clustering result back into the brain image space in Figure 8.1. The black areas in Figure 8.1 are places where the time series exhibit no temporal dependence and are therefore excluded before clustering. The yellow voxels constitute the second cluster, which is largely located in the visual cortex. The purple cluster covers most of the rest of the grey matter of the brain. Therefore simply from these plots in Figure 8.1, the clustering detects the visual cortex which is expected to be active over the course of the experiment, since the subject is asked to gaze at a fixation point. Therefore the signals in the visual cortex should be different from other areas. Our algorithm has detected this difference. Another interesting

fact is that this spacial separation is done based solely on the temporal dependence in the fMRI signals. Information about the signal magnitude, spacial correlation are not used in the clustering.

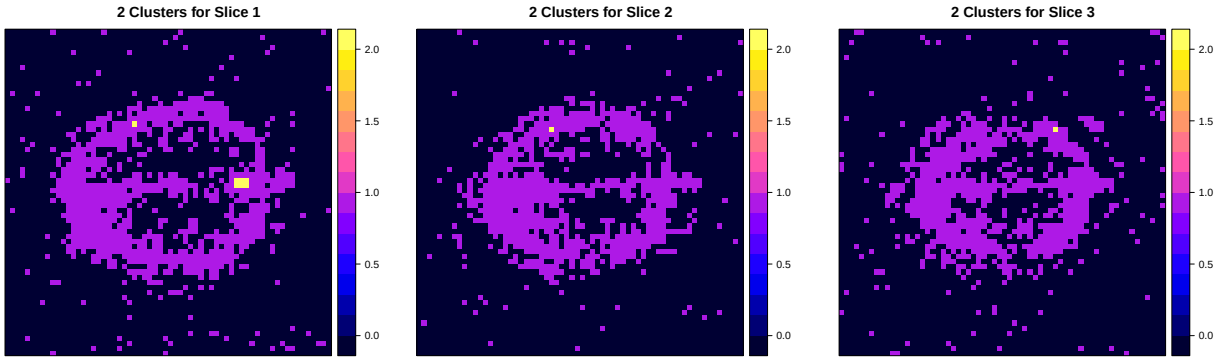


(a) (b) (c)

Figure 8.1: The two clusters detected using CHOMP(1)-based clustering

Additionally, Figure 8.1 shows that the detected visual region is dissipating from the inferior slice to the superior slice. This observation matches with the fact that the horizontal location (frontal sinus) of the inferior slice is at the upper edge of the visual cortex. This dissipating observation is more obvious in Figure 8.2 after we remove some abnormal spikes in the signal (time points from 660 to 730, and 1000 to 1070 on all series). The removal is suggested in an independent observation by the psychologists in the fMRI research group at the University of Georgia.

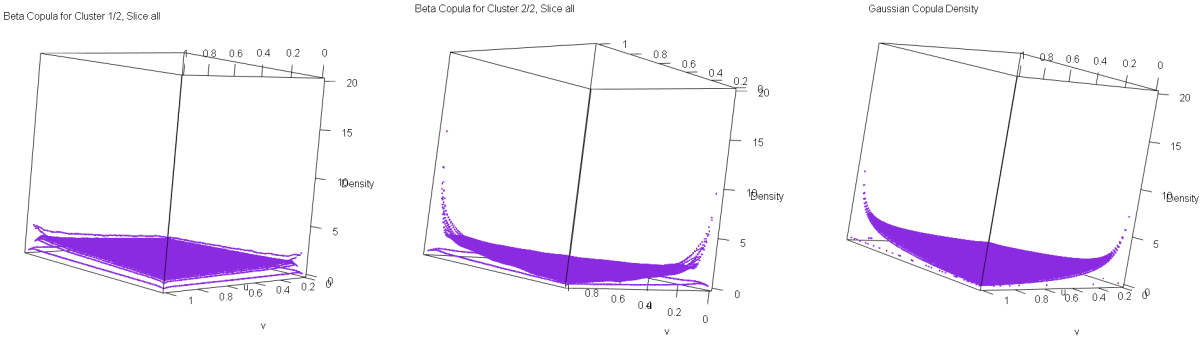
Another thing one may want to see is what kind of temporal dependence are displayed in the two clusters, and how different they are. We further look into the two clusters in Figure 8.1 by plotting their estimated copula functions which fully determine their temporal dependence. We plot the estimated copula densities in Figure 8.3a and Figure 8.3b. From the plots, we can see that the time series in cluster 1 has very little dependence, since the copula density is mostly flat. With such weak dependence, the dependence structure doesn't



(a) (b) (c)

Figure 8.2: The two clusters detected using CHOMP(1)-based clustering after removing additional outliers

matter a lot because the data will be very close to independent. In Figure 8.3b, the temporal dependence for Cluster 2 is much stronger as the copula density gradually gets higher when approaching the two corners,(0,0) and (1,1) which also indicates the dependence is positive.



(a) Cluster 1 (b) Cluster 2 (c) Gaussian Copula

Figure 8.3: A comparison of the clusters' estimated copula density functions

Moreover, we want to see the dependence structure for Cluster 2 since capturing the structure of temporal dependence is one of the most important advantage of our algorithm. The dependence structure is very hard to describe. But we can have a general idea by a simple

comparison. We compare the dependence structure of Cluster 2 in Figure 8.3b with a linear dependence structure. We display the linear dependence structure by plotting a Gaussian copula density in Figure 8.3c. The Gaussian copula density we use is set to have the same dependence strength as Cluster 2, so we can focus just on the structure. Comparing the two copula densities in Figure 8.3b and Figure 8.3c, we can see that: (1) the copula density for Cluster 2 has higher tails than the Gaussian copula in the two corners. This means Cluster 2 may very likely exhibit both upper and lower tail dependence. (2) the copula function for Cluster 2 is slightly asymmetric because the tail in the (1,1) corner is slightly higher than the one in the (0,0) corner. Therefore the upper tail dependence is slightly stronger than the lower tail dependence..

But what does the observed dependence structure mean for the fMRI signals in Cluster 2? As discussed in Section 5.2.3, tail dependence in the copula will lead to longer range temporal dependence in time series generated by the CHOMP. This is because when an extreme fMRI signal occurs, its effect to later signals becomes stronger than usual due to the tail dependence. This stronger dependence makes the effect of that extreme signal be carried over for a longer time, hence results in the longer range dependence. If the upper tail dependence is stronger than the lower tail dependence, the longer range dependence will occur more often for extremely high signals than that for extremely low signals. After we consult with neuroscience experts, this longer range dependence does make practical sense. This is because the visual cortex is kept active for a long time since subject keeps staring at a fixation point in the experiment.

After all, the observations from Figure 8.3 indicate that the dependence structure in Cluster 2 is not at all linear. Therefore the CHOMP model used in our algorithm is more suitable than models that only allow linear dependence. This is also why our clustering algorithm using CHOMP performs better than the AR(1)-based clustering algorithm.

Chapter 9

Discussion and Future Work

In this dissertation, we have discussed model-based time series clustering. The time series we consider here are discrete time Markov chains with real state space. In Chapter 2, we introduce the motivation of this work. That is, using more information from the data should make clustering better and easier. This motivates us to develop new methods for model-based time series clustering because previous methods in the literature use very limited information for their clustering. A literature review on model-based time series clustering is given in Chapter 3.

We propose using the copula based higher order Markov process (CHOMP) by Ibragimov (2009) in time series clustering. It has several advantages over other time series models used in the literature: (1) It is able to capture not only the temporal dependence strength, but also the dependence structure in the time series data. (2) It allows both linear and non-linear temporal dependence. In Chapter 5, we first generalize Ibragimov's result to a more general framework so that methods other than copulas can also be used to characterize a higher order Markov chain (HOMP). But more important is that we relax the stationarity

condition in the original version of CHOMP by Ibragimov (2009) in Corollary 5.7 so that the modified CHOMP can also be used for non-stationary time series with arbitrary profiles. In other words, the modified CHOMP can also capture the profile information of the time series. In the same Chapter, we have discussed both the semi-parametric CHOMP and the non-parametric CHOMP. The semi-parametric CHOMP is limited by the parametric form of the copula function, so we propose a non-parametric CHOMP and give a detailed estimation methods based on the two-step procedure by Chen and Fan (2006).

In Chapter 6, we propose our model-based time series clustering algorithm by combining the CHOMP model with agglomerative hierarchical clustering and the finite mixture model. The former is for clustering stationary time series, and the latter is for clustering non-stationary time series. A method to determine the number of clusters is also proposed for each of them. In Chapter 7 and Chapter 8, we conduct an extensive simulation study and two real data analyses and compare our algorithm with k-means clustering and AR(1)-based clustering algorithm. Other models like ARIMA and dynamic regression models are essentially the same as the AR model in terms of modeling the temporal dependence because they all only allow linear dependence. In the simulations and real data analyses, we find that our proposed algorithm outperforms its competitors in most situations due to its ability to capture more information from the data for clustering. In other situations, our algorithm performs equally well with its competitors. These findings further demonstrate that with more useful information available, clustering becomes better and easier.

However, there is still limitation in our method. We propose a non-parametric CHOMP and give a non-parametric estimation method based on the two-step procedure by Chen and Fan (2006). The theoretical property of this estimator needs to be established. In fact the consistency of the non-parametric estimator should not be hard to prove. We can follow the same idea of the proof given by Chen and Fan (2006) for the semi-parametric

two-step estimation. One difference is now we use non-parametric copula estimators instead of the MLE for the parametric copula. But the consistency of most non-parametric copula estimators we use are already given by Geenens et al. (2014). Therefore combining the work of Chen and Fan (2006) and Geenens et al. (2014) should lead to the desired result.

On the other hand, we only consider using non-parametric CHOMP with first order Markovian dependence in our proposed algorithm. This is because estimating a non-parametric CHOMP(q) involves estimating a non-parametric copula that is $(q+1)$ dimensional. When $q \geq 2$, one will need to estimate a multivariate copula function non-parametrically. This is very difficult due to the notorious "curse of dimensionality" and the huge amount of data needed. However, the vine copula might be a promising method to estimate $(q+1)$-dimensional copula functions when $q > 2$. The idea originates from Joe (1996) and is further developed by Bedford and Cooke (2002) and Kurowicka and Cooke (2006). The recent breakthrough is due to Aas et al. (2009), and a R package called $CDVine$ is also developed by Brechmann and Schepsmeier (2013). The basic spirit of the vine copula is the same splitting idea in the original copula theory. Since now we consider a high dimensional copula function, it is very hard to directly model the entire copula. Therefore one can split it into blocks each of which only considers two dimensions, then these small blocks are combined to reconstruct the original high dimensional copula function. Specifically, let us consider a three dimensional case. Let $(X_1, X_2, X_3) \sim F$ with marginal c.d.f. F_1, F_2, F_3, and $f(x_1, x_2, x_3)$, f_1, f_2, f_3 be the corresponding densities, then with recursive conditioning we can have

$$f(x_1, x_2, x_3) = f_1(x_1) f(x_2|x_1) f(x_3|x_1, x_2) \tag{9.0.1}$$

In Equation 9.0.1, $f(x_2|x_1)$ and $f(x_3|x_1, x_2)$ can be written as

$$f(x_2|x_1) = \frac{f(x_1, x_2)}{f(x_1)} = \frac{c_{1,2}(F_1(x_1), F_2(x_2))f_1(x_1)f_2(x_2)}{f_1(x_1)} = c_{1,2}(F_1(x_1), F_2(x_2))f_2(x_2) \tag{9.0.2}$$

and

$$\begin{aligned} f(x_3|x_1, x_2) = \frac{f(x_2, x_3|x_1)}{f(x_2|x_1)} &= \frac{c_{2,3|1}(F(x_2|x_1), F(x_3|x_1))f(x_2|x_1)f(x_3|x_1)}{f(x_2|x_1)} \\ &= c_{2,3|1}(F(x_2|x_1), F(x_3|x_1))f(x_3|x_1) \\ &\overset{(3)}{=} c_{2,3|1}(F(x_2|x_1), F(x_3|x_1))c_{1,3}(F_1(x_1), F_3(x_3))f_3(x_3) \end{aligned}$$

Therefore the three dimensional density in Equation 9.0.1 can be represented by bivariate copula densities $c_{1,2}$, $c_{1,3}$ and $c_{2,3|1}$. These copula densities are called the pair copulas and can be sequentially estimated. Typically the conditional copula density $c_{2,3|1}$ is assumed to be independent of the conditioning variable X_1. For more details, see Aas et al. (2009) and Brechmann and Schepsmeier (2013).

In the simulation study and two real data analyses, we compare our algorithm using CHOMP with k-means clustering and the AR(1)-based clustering algorithm. One may argue why not compare with other model-based algorithms. One reason is that most of the time series models used only allow for linear temporal dependence, including ARIMA and dynamic regression models. Therefore from the perspective of temporal dependence, these models are no different from AR. However, one class of time series model we didn't explore here is the non-parametric time series models, although there is not much work done in time series clustering using non-parametric models. In fact, the CHOMP model we use is a non-parametric time series model. But it is worth comparing our algorithm using CHOMP with

algorithms using other non-parametric time series models.

Another point that is worth discussing is the order of dependence in the time series. In our work, we assume they all have the same order of Markovian property, but it is possible that data with different orders need to be clustered. In that case, one solution is to estimate the orders beforehand, and cluster the time series by order first, and then re-cluster within each sub-cluster. One simple method to estimate the orders is through modeling AR(p) with different values of p, and see which one gives the highest AIC/BIC. However, this only considers linear dependence. Therefore it might be better to use certain conditional independence tests that does not assume any distributional or linear assumption.

The idea we keep repeating in this work is: with more useful information available, the clustering becomes easier and better. Our algorithm using the CHOMP does provide that in the context of model-based time series clustering. However, another question we may ask is: do we need all information to cluster? If not, how much is enough? This is a very difficult question to answer. Recall the apple and orange example, the texture or the color information is good enough to cluster them into two groups. However, if we do not know they are apples and oranges, how can we know that only texture or color is good enough. Moreover, what if there are actually three clusters: apple, orange and green orange? Then with only texture, we are not able to detect the third cluster. Of course, that also depends on how deep one wants to go in clustering. Someone may be good with two clusters, but someone may prefer to have three clusters with the green orange cluster detected. All of these uncertainties make it very hard to answer how much information is enough for clustering. If we don't know the answer, why not have as much information as possible? Then based on the complete information, it will be easier to determine what is useful and what is not.

Chapter 10

Bibliography

Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009), "Pair-copula constructions of multiple dependence. Insurance: Mathematics and economics," **44**(2), 182-198.

Akaike, H. (1973), "Information theory and an extension of the maximum likelihood principle," In: Petrov, B.N., Csaki, F. (Eds.), 2nd International Symposium on Information Theory. Akademiai Kiado, Budapest, 267-281.

Autin, F., Lepennec, E. and Tribouley, K. (2010), "Thresholding methods to estimate the copula density," *Journal of Multivariate Analysis*, **101**, 200-222.

Bailey, K. (1994). "Numerical Taxonomy and Cluster Analysis," *Typologies and Taxonomies*, page 34.

Baker, F. B. and Hubert, L. J. (1975), "Measuring the power of hierarchical cluster analysis," *Journal of the American Statistical Association*, **70**, 31-38.

Banfield, J. D. and Raftery, A. E. (1993), "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, **49**, 803-821.

Baragona, R. (2001), "A simulation study on clustering time series with meta-heuristic methods," *Quaderni di Statistica*, **3**, 1-26.

Bauwens, L. and Rombouts, J. V. (2007), "Bayesian clustering of many GARCH models," *Econometric Reviews*, **26**(2-4), 365-386.

Bedford, T. and Cooke, R.M. (2002), "Vinesa new graphical model for dependent random variables," *Annals of Statistics*, **30**, 1031-1068.

Berchtold, A. and Raftery, A. E. (2002), "The mixture transition distribution model for high-order Markov chains and non-Gaussian time series," *Statistical Science*, **17**(3), 328-356.

Biernacki, C., Celeux, G. and Govaert, G. (2000), "Assessing a mixture model for clustering with the integrated completed likelihood," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(7), 719-725.

Bowman, A., Hall, P., and Prvan, T. (1998), "Bandwidth selection for the smoothing of distribution functions," *Biometrika*, **85**(4), 799-808.

Bozdogan, H. (1987), "Model selection and Akaikes information criterion (AIC): the general theory and its analytical extensions," *Psychometrika*, **52** (3), 345-370.

Brechmann, E. C. and Schepsmeier, U. (2013), "Modeling dependence with c- and d-vine copulas: The R package CDVine," *Journal of Statistical Software*, **52**(3), 1-27.

Brock, G., Pihur, V., Datta, S. and Datta, S. (2008), "clValid: an R package for cluster validation," *Journal of Statistical Software*, **25**(4).

Cadez, I.V., Gaffney, S. and Smyth, P. (2000a), "A general probabilistic framework for clustering individuals and objects," In *Proceedings of the Sixth ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, 140-149, Boston, MA, USA, 20-23 August.

Cadez, I., Heckerman, D., Meek, C., Smyth, P. and White, S. (March 2000b, Revised September 2001), "Model-based clustering and visualization of navigation patterns on a web site," Technical Report, MSR-TR-00-18, Microsoft Research, Redmond, WA, USA.

Cambanis, S., Huang, S. and Simons, G. (1981), "On the theory of elliptically contoured distributions," *Journal of Multivariate Analysis*, **11**, 368-385.

Cadez, I., Heckerman, D., Meek, C., Smyth, P. and White, S. (2000c), "Visualization of navigation patterns on a web site using model-based clustering," In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 280-284, Boston, MA, USA, 20-23 August.

Celeux, G. and Govaert, G. (1992), "A classification EM algorithm for clustering and two stochastic versions," *Computational statistics and Data analysis*, **14**(3), 315-332.

Charpentier, A., Fermanian, J. D., and Scaillet, O. (2007). "The estimation of copulas: Theory and practice," *Copulas: from theory to applications in finance*, 35-62.

Cheeseman, P. and Stutz, J. (1995), "Bayesian classification (AutoClass): Theory and results," in *Advances in Knowledge Discovery and Data Mining*, eds. U. Fayyad, G. Piatesky-Shapiro, P. Smyth and R. Uthurusamy, AAAI Press, Menlo Park, CA, 153-180.

Chen, S. X. (1999), "Beta kernel estimators for density functions," *Computational Statistics & Data Analysis*, **31**, 131-145.

Chen, X. and Fan, Y. (2006), "Estimation of copula-based semiparametric time series models," *Journal of Econometrics*, **130**, 307-335.

Chen, S. X., and Huang , T.-M. (2007), "Nonparametric estimation of copula functions for dependence modelling," *Canaddian Journal of Statistics*, **35**, 265-282.

Chen, X., Wu, W. B., and Yi, Y. P. (2009), "Efficient estimation of copula-based semiparametric Markov models," *The Annals of Statistics*, **37**, 4214-4253.

Ching, W. K., Fung, E. S. and Ng, M. K. (2004), "Higher-order Markov chain models for categorical data sequences," *Naval Research Logistics*, **51**, 557-574.

Clayton, D. G. (1978), "A model for association in bivariate life tables and its application in epidemiological studies of familial tendency in chronic disease incidence," *Biometrika*, **65**, 141-151.

Dawson, K. J. and Belkhir, K. (2009), "An agglomerative hierarchical approach to visualization in Bayesian clustering problems," *Heredity (Edinb)*, **103**, 32-45.

Deheuvels, P. (1979), "La fonction de dépendance empirique et ses propriétés. Un test non paramériqued'indépendance." *Académie Royale de Belgique. Bulletin de la Classe des Sciences. 6e Série.*, **65**, 274-292.

Diday, E., Brito, P. and Mfoumoune, E. (1993), "Learning probabilistic models by conceptual pyramidal clustering," *Lecture Notes in Computer Science*, **727**, 362.

Duong, T. and Hazelton, M.L. (2003), "Plug-in bandwidth matrices for bivariate kernel density estimation," *Journal of Nonparametric Statistics*, **15**, 17-30.

Duong, Tarn (2014), "Data-based smoothing for non-parametric estimation of copula functions and their densities," *submitted*.

Dunn, J.C. (1973), "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, **3**, 32-57.

Embrechts, P., Lindskog, F. and McNeil, A. (2003), "Modelling dependence with copulas and applications to risk management," *Handbook of heavy tailed distributions in finance*, **8**(1), 329-384.

Epanechnikov, V. A. (1969), "Non-parametric estimation of a multivariate probability density," *Theory of Probability and Its Applications*, **14**(1), 153-158.

Fang, K.-T., Kotz, S. and Ng, K.-W. (1987), *Symmetric Multivariate and Related Distributions*, Chapman and Hall, London.

Ferguson, T.S. (1973), "A Bayesian analysis of some nonparametric problems," *Annals of Statistics*, **1**, 209-230.

Fermanian, J.-D., Radulovič, D. and Wegkamp, M. (2004), "Weak convergence of empirical copula processes," *Bernoulli*, **10**, 847-860.

Florek, K., Lukaszewicz, J., Perkal, J. and Zubrzycki, S. (1951a), "Sur la liaison et la division des points d'un ensemble fini," *Colloquium Mathematicae*, **2**, 282-285.

Florek, K., Lukaszewicz, J., Perkal, J. and Zubrzycki, S. (1951b), "Taksonomia Wroclawska," *Przeglad Antropol.*, **17**, 193-211.

Fraley, C. and Raftery, A. E. (2002), "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, **97**, 611-631.

Frank, M.J. (1979), "On the simultaneous associativity of $F(x, y)$ and $x + y - F(x, y)$," *Aequationes Math*, **19**, 194-226.

Frühwirth-Schnatter, S. and S. Kaufmann (2008), "Model-based clustering of multiple time series," *Journal of Business and Economic Statistics*, **26**, 78-89.

Frühwirth-Schnatter, S. (2011), "Model-based Clustering of Time Series-A Review from a

Bayesian Perspective".

Gänssler, P. and Stute, W. (1987), *Seminar on Empirical Processes*, Birkhauser Boston Inc..

Geenens, G., Charpentier, A. and Paindaveine, D. (2014) "Probit transformation for non-parametric kernel estimation of the copula density", *submitted*.

Genest, C., Masiello, E. and Tribouley, K. (2009), "Estimating copula densities through wavelets," *Insurance: Mathematics and Economics*, **44**, 170-181.

Gijbels, I. and Mielniczuk, J. (1990), "Estimating the density of a copula function," *Communications in Statistics-Theory and Methods*, **19**(2), 445-464.

Gumbel, E.J. (1960), "Distributions des valeurs extrêmes en plusiers dimensions," *Publications de l'Institut de Statistique de l'Université de Paris*, **9**, 171-173.

Heard, N. A., Holmes, C. C. and Stephens, D. A. (2006), "A quantative study of gene regula- tion involved in the immune response of anopheline mosquitoes: an application of Bayesian hierarchical clustering of curves," *Journal of the American Statistical Association*, **101**, 18-29.

Herrero, J., Valencia, A. and Dopazo, J. (2001), "A hierarchical unsupervised growing neural network for clustering gene expression patterns," *Bioinformatics*, **17**(2), 126-136.

Hofert, M., Maechler, M. and McNeil, A. J. (2012), "Estimators for Archimedean copulas in high dimensions," arXiv:1207.1708v2.

Hjort,N. L., Holmes, C., Müller, P. and Walker, S. G. (2010), *Bayesian Nonparametrics*, Cambridge University Press.

Hurd, M., Salmon, M. and Schleicher, C. (2005), "Using copulas to construct bivariate

foreign exchange distributions with an application to the sterling exchange rate index," CEPR Discussion Paper No. 5114.

Hutchinson, T. P. and Lai, C. D. (1990), *Continuous Bivariate Distributions, Emphasising Applications*, Rumsby Scientific Publishing, Adelaide.

Ibragimov, R. (2009), "Copula-based characterizations for higher-order Markov processes," *Econometric Theory*, **25**, 819-846.

Izenman, A. J. (1991), "Recent developments in nonparametric density estimation," *Journal of the American Statistical Association*, **86**, 205-224.

Joe, H. (1993), "Parametric families of multivariate distributions with given margins," *Journal of Multivariate Analysis*, **46**, 262-282.

Joe, H. (1996), "Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters," *Lecture Notes-Monograph Series*, 120-141.

Joe, H. (1997), *Multivariate models and multivariate dependence concepts*, CRC Press.

Jones, M. C. (1993), "Simple boundary correction for kernel density estimation," *Statistics and Computing*, **3**(3), 135-146.

Juárez, M. A. and Steel, M. F. J. (2010), "Model-based clustering of non-Gaussian panel data based on skew-t distributions," *Journal of Business and Economic Statistics*, **28**, 52-66.

Kalpakis, K., Gada, D. and Puttagunta, V. (2001), "Distance measures for effective clustering of ARIMA time-series," *Proceedings of the 2001 IEEE International Conference on Data Mining*, San Jose, CA, November 29-December 2, 273-280.

Kaufman, L. and Rousseeuw, P.J. (1987), "Clustering by means of medoids," In *statistical*

data analysis based on the L_1norm and related methods, ed. Y. Dodge, North-Holland, 405-416.

Kendall, M. (1938), "A new measure of rank correlation," *Biometrika*, **30**, 81-89.

Kurowicka, D. and Cooke, R. M. (2006), *Uncertainty analysis with high dimensional dependence modelling*, John Wiley and Sons.

Kullback, S. and Leibler, R.A. (1951), "On information and sufficiency," *Annals of Mathematical Statistics*, **22**, 79-86.

Krogh, A., Brown, M., Mian, I.S., Sjölander, K. and Haussler, D. (1994), "Hidden Markov models in computational-biology applications to protein modeling," *Journal of Molecular Biology*, **235**, 1501-1531.

Law, M.H. and Kwok, J.T. (2000), "Rival penalized competitive learning for model-based sequence clustering," In *Proceedings of the Fifteenth International Conference on Pattern Recognition*, **2**, 195-198, Barcelona, Spain, 3-7th September.

Lee, M.-L. T. (1996), "Properties and applications of the Sarmanov family of bivariate distributions," *Communications in Statistics: Theory and Methods*, **25**, 1207-1222.

Lejeune, M. and Sarda, P. (1992), "Smooth estimators of distribution and density functions," *Computational Statistics and Data Analysis*, **14**(4), 457-471.

Lentzas, G. and Ibragimov, R. (2008), Copulas and long memory, Harvard Institute of Economic Research Discussion Paper No. 2160.

Li, X., Mikusinski, P. and Taylor, M. D. (1998), "Strong approximations of copulas," *Journal of Mathematical Analysis and Applications*, **225**, 608-623.

Li, C. and Biswas, G. (1999), "Temporal pattern generation using hidden Markov model

based unsupervised classification," in: D.J. Hand, J.N. Kok, M.R. Berthold (Eds.), *Lecture Notes in Computer Science*, **164**, 245-256, IDA, Springer, Berlin.

Li, C. and Biswas, G. (2000), "A Bayesian approach to temporal data clustering using hidden Markov models," In *Proceedings of the Seventeenth International Conference on Machine Learning*, 543-550, Stanford, CA, USA, 29 June - 2 July.

Liao, T. Warren (2005), "Clustering of time series data-a survey," *Pattern Recognition*, **38**, 1857-1874.

Liao, T. Warren (2007), "A clustering procedure for exploratory mining of vector time series," *Pattern Recognition*, **40**(9), 2550-2562.

Lindquist, M. A. (2008), "The statistical analysis of fMRI data", *Statistical Science*, **23**(4), 439-464.

Liu, Y., Hayes, D. N., Nobel, A. and Marron, J. S. (2008), "Statistical significance of clustering for high-dimensional, low sample size data," *Journal of the American Statistical Association*, **103**, 1281-1293.

Loader, C.R. (1996), "Local likelihood density estimation," *Annals of Statistics*, **24**, 1602-1618.

Lopez-Paz, D., Hernández-Lobato, J.M. and Schölkopf, B. (2013), "Semi-supervised domain adaptation with non-parametric copulas", *Manuscript*.

Lowin, Jeremiah L. (2010), "The Fourier Copula: Theory & Applications," *Available at SSRN 1804664*.

Luan, Y. and Li, H. (2003), "Clustering of time-course gene expression data using a mixed-effects models with B-splines," Bioinformatics, **19**, 474-482.

MacNaughton-Smith, P., Williams, W.T. and Mockett, L.G. (1964), "Dissimilarity analysis: A new technique of hierarchical subdivision," *Nature*, **202**, 1034-1035.

MacQueen, J. (1967), "Some methods for classification and analysis of multivariate observations," In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, **1**, 281-297, eds. L. M. L. Cam and J. Neyman, Berkeley, CA, University of California Press.

Maharaj, E.A. (2000), "Clusters of time series," *Journal of Classification*, **17**, 297-314.

McLachlan, G. J. and Basford, K. E. (1988), *Mixture Models: Inference and Applications to Clustering*, New York, Marcel Dekker.

McShane, L. M., Radmacher,M. D., Freidlin, B., Yu, R., Li, M.-C. and Simon, R. (2002), "Methods for assessing reproducibility of clustering patterns observed in analyses of microarray data," *Bioinformatics*, **18**, 1462-1469.

Maitra, R., and Melnykov,V. (2010a), "Assessing significance in finite mixture models," Technical Report, Department of Statistics, Iowa State University.

Maitra, R., Melnykov, V. and Lahiri, S. N. (2012), "Bootstrapping for significance of compact clusters in multidimensional datasets," *Journal of the American Statistical Association*, **107**, 378-392.

Medvedovic, M., Yeung, K. Y. and Bumgarner, R. E. (2004), "Bayesian mixture model based clustering of replicated microarray data," *Bioinformatics*, **20**(8), 1222-1232.

Morettin, P. A., Toloi, C. M.C., Chiann, C. and Miranda, J. C.S. de (2011), "Wavelet Estimation of Copulas for Time Series," *Journal of Time Series Econometrics*, **3**(3), article 4.

Nelsen, R. B. (2006). *An Introduction to Copulas* (Second Edition), New York, NY, Springer Science & Business Media Inc.

Nieto-Barajas, L. E., and Contreras-Cristán, A. (2014), "A Bayesian nonparametric approach for time series clustering," *Bayesian Analysis*, **9**(1), 147-170.

Olkin, I. and Liu, R. (2003), "A bivariate beta distribution," *Statistics and Probability Letters*, **62**, 407-412.

Omelka, M., Gijbels, I. and Veraverbeke, N. (2009). "Improved kernel estimation of copulas: Weak convergence and goodness-of-fit testing," *Annals of Statistics*, **37**(5B), 3023-3058.

Owsley, L.M.D., Atlas, L.E. and Bernard, G.D. (1997), "Self-organizing feature maps and hidden Markov models for machine-tool monitoring," *IEEE Transactions on Signal Processing*, **45**, 2787-2798.

Park, C., Lazar, N. A., Ahn, J. and Sornborger, A. (2010), "A multiscale analysis of the temporal characteristics of resting-state fMRI data," *Journal of neuroscience methods*, **193**(2), 334-342.

Pena, J. M., Lozano, J. A., and Larranaga, P. (1999), "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern recognition letters*, **20**(10), 1027-1040.

Piccolo, D. (1990), "A distance measure for classifying ARMA models," *Journal of Time Series Analysis*, **11**, 153-163.

Poulsen, C. S. (1990), "Mixed Markov and latent Markov modelling applied to brand choice behaviour," *International Journal of Research in Marketing*, **7**, 5-19.

Perrone, M. P. and Connell, S. P. (2000), "K-means clustering for hidden Markov mod-

els," In *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, pp. 229-238, Amsterdam, Netherlands, 11-13 September.

Raftery, A. E. (1985), "A model for higher-order Markov chain," *Journal of the Royal Statistical Society. Series B(Methodological)*, **47**(3), 528-539.

Ramoni, M., Sebastiani, P. and Cohen, P. (2002), "Bayesian clustering by dynamics," *Machine Learning*, **47**, 91-121.

Ridgeway, G. (1997), "Finite discrete Markov process clustering," Technical Report, MSR-TR-97-24, Microsoft Research, Redmond, WA, USA, September.

Sancetta, A. and Satchell, S. (2004), "The bernstein copula and its applications to modeling and approximations of multivariate distributions," *Econometric Theory*, **20**(3),535-562.

Schwarz, G. (1978), "Estimating the dimension of a model," *The Annals of Statistics*, **6** (2), 461-464.

Schweizer, B. (1991), "Thirty years of copulas," *Advances in Probability Distributions with Given Marginals* (Volume 67), ed. DallAglio, G., Kotz, S. and Salinetti, G., Dordrecht, Springer Netherlands, 13-50.

Sheather, S. J. and Jones, M. C. (1991), "A Reliable Data-Based Bandwidth Selection Method for Kernel Density Estimation," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 53, No. 3, 683-690.

Sklar, A. (1959), "Fonctions de répartition á n dimensions et leurs marges," *Publications de l'Institut de Statistique de l'Université de Paris*, **8**, 229-231.

Smyth, P. (1997), "Clustering sequences with hidden Markov models," In *Advances in Neural Information Processing Systems*, **9**, MIT Press, 648-654.

Smyth, P. (1999), "Probabilistic model-based clustering of multivariate and sequential data," In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, 299-304, Fort Lauderdale, FL, USA, 4-6 January.

Stephens, M. (2000), "Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods," *Annals of Statistics*, 40-74.

Tsukahara, H. (2005), "Semiparametric estimation in copula models," *Canadian Journal of Statistics*, **33**(3), 357-375.

Tucker, H. G. (1959), "A generalisation of the Glivenko-Canttelli theorem," *The Annals of Mathematical Statistics*, **30**, 820-830.

Vrac, M., Billard, L., Diday, E. and Chédin, A. (2012), "Copula analysis of mixture models," *Computational Statistics*, **27**, 427-457.

Wilpon, J.G. and Rabiner,L.R. (1985), "Modified k-means clustering algorithm for use in isolated word recognition," *IEEE Transactions on Signal Process*, **33** (3), 587-594.

Xiong,Y. and Yeung, D.-Y. (2002), "Mixtures of ARMA models for model-based time series clustering," *Proceedings of the IEEE International Conference on Data Mining*, Maebaghi City, Japan, 9-12 December.