Kernel-based Copula Processes

by

Eddie K. H. Ng

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Electrical & Computer Engineering
University of Toronto

Abstract

Kernel-based Copula Processes

Eddie K. H. Ng

Doctor of Philosophy

Graduate Department of Electrical & Computer Engineering

University of Toronto

2010

The field of time-series analysis has made important contributions to a wide spectrum of applications such as tide-level studies in hydrology, natural resource prospecting in geo-statistics, speech recognition, weather forecasting, financial trading, and economic forecasts and analysis. Nevertheless, the analysis of the non-Gaussian and non-stationary features of time-series remains challenging for the current state-of-art models.

This thesis proposes an innovative framework that leverages the theory of copula, combined with a probabilistic framework from the machine learning community, to produce a versatile tool for multiple time-series analysis. I coined this new model *Kernel-based Copula Processes* (KCPs). Under the new proposed framework, various idiosyncracies can be modeled compactly via a *kernel function* for each individual time-series, and long-range dependency can be captured by a copula function. The copula function separates the marginal behavior and serial dependency structures, thus allowing them to be modeled separately and with much greater flexibility. Moreover, the codependent structure of a large number of time-series with potentially vastly different characteristics can be captured in a compact and elegant fashion through the notion of a *binding copula*. This feature allows a highly heterogeneous model to be built, breaking free from the homogeneous limitation of most conventional models. The KCPs have demonstrated superior predictive power when used to forecast a multitude of data sets from meteorological and financial areas. Finally, the versatility of the KCP model is exemplified when it was successfully applied to non-trivial classification problems unaltered.

Acknowledgements

First, I would like to express my most sincere gratitude to my thesis supervisor, Sebastian Jaimungal, for a great journey of research together. I am indebted to him for his attentive guidance and support given at every step of the way. I have enjoyed all our research meetings together. His energy is contagious and his creativity is inspiring. I have especially enjoyed his insightful post-talk commentaries and interpretations at conferences, which always made everything clear and instantly sensible.

I would also like to thank my co-supervisor Brendan Frey for guiding me through the earlier part of my Ph.D. studies and for his continued support throughout my Ph.D.

My thanks also go to Frank Kschischang for many stimulating conversations and for encouraging me to pursue my research interests.

I would like to remember Sam Roweis. I am fortunate to have him as a first teacher on many topics in machine learning. He was truly a great educator and human being. It is tragic that he is no longer with us and that future generations will not have the opportunity to benefit from his teachings.

I gratefully acknowledge the Natural Sciences and Engineering Council of Canada and the Government of Ontario for their funding support.

I am grateful for all the new friends I made throughout my Ph.D. studies, who have made my transition back to school and the hectic life that goes with it more enjoyable.

I would also like to thank my long-time friends: Philip and Kason for all the weekend jogs; Lawrence for all the great hikes; Wendy for listening to my rants; and Kim for being a great friend and keeping me sane all these years.

Finally, I thank my mom, sister, and brother for their love and support, and for always believing in ability to succeed. Special thanks go to my better-half, Sandy, for without her patience and support, none of this would be possible. I am especially impressed by and grateful for her persistence in proof-reading my thesis, no matter how many times it put her to sleep.

Contents

List of Symbols and Acronyms

D	A scalar typically denotes the number dimensionality or the number of data series in a data set.
K	Gram matrices, generated by evaluating kernel functions at some input coordinates $\{x\}_{i=1}^{N} \times \{x\}_{i=1}^{N}$
M	A scalar typically denotes the number of data points in the test set.
N	A scalar typically denotes the number of data points in the training set.
Σ or Λ	Covariance or Gram matrices.
$\mathbb{C}(\cdot)$	Copula functions.
$\mathbb{E}(\cdot)$	Expectation over a certain probability measure.
$\mathbb{P}(Y = y)$ or $f(y)$	Probability density functions.
$\mathbb{P}(Y \leq y)$ or $F(y)$	Probability distribution functions.
$\mathbf{X}, \mathbf{Y}, \mathbf{W}$	Bold and capitalized letters typically denotes matrix variables.

$\mathbf{x}, \mathbf{y}$	Vectored valued input variable and targets. In some cases, the bold typeface can also denotes a collection of input variables and targets $\{x\}_{i=1}^{N}, \{y\}_{i=1}^{N}$.
$\mathfrak{c}(\cdot)$	Copula density functions.
ν	Degree of freedom of a Student's t distribution or copula function.
$k(x_i, x_j)$	Kernel functions.
u	Typically denotes a uniformly distributed random variable, often as a result of a target variable undergone transformation by the marginal distribution function, $u = F(y)$.
x, y	Scalar input variables (e.g. time) and targets respectively.
z	Typically denotes a random variable transformed by the inverse distribution function as part of an elliptical copula function, e.g. , $z = \Phi^{-1}(F(y))$.
ADC	Anderson-Darling Criterion
AIC	Akaike Information Criterion
ARMA	Autoregressive Moving Average model
BIC	Bayesian Information Criterion
DGP	Data Generating Process
GPC	Gaussian Process Classification
GPs	Gaussian Processes
HMM	Hidden Markov Model

KCPs	Kernel-based Copula Processes
LDSs	Linear Dynamic Systems
PITs	Probability Integral Transforms
WGPs	Warped Gaussian Processes

Chapter 1

Introduction

Time series analysis can be broadly classified into two main tasks: (1) general regression, forecasting or prediction, and (2) classification. General regression can be used for filling in missing data and general curve fitting. Regression can be loosely thought of as *prediction* given values from the future and the past; similarly, forecasting is regression with only information given from the past. Classification allows database query and retrieval of time-series.

Over the years, time series analysis has become deeply intertwined with many different disciplines, playing an ever more vital role in our lives. The most noted example is in the area of economics and finance, where an entire field has sprung up in the name of *financial econometrics* [42] [47]. This field has provided a vast intellectual playground for academics and practitioners. The variety and volume of data available is enormous. Millions of new data points are being generated everyday, from the relatively tame but structure-rich bond prices and interest rates data to the volatile commodity prices such as sugar and crude oil prices, and of course, the notoriously spiky electricity prices. The recent trend of migrating to high-frequency data, where details from each transaction are being recorded and analyzed, have increased the demands for newer and faster models and algorithms. Forecasting is the name of the game for econometrics. Policy-makers

routinely consult economic forecasts in setting fiscal and monetary policies. Power plant operators forecast electricity demands to choose the types and operating levels of power plants. Time series forecasting is also central to fields such as meteorology, where the ability to forecast temperature and the dynamics of weather systems accurately could mean significant economic and social benefits or even serves as the input to early warning systems. The demand for predictions with longer time-frame and higher accuracy continues to grow.

Independently, researchers and practitioners from other fields have also made significant progress towards better forecasting tools. For instance, the Kalman filter [57] represents one such success from the field of engineering. It was at the heart of the on-board navigation system for the Apollo mission [16]. Since then, it has permeated to a great number of other applications including navigation in global positioning systems (GPS) [66] and human motion tracking [104]. Finally, Kalman filter was extended to finance and econometrics where time-varying extensions of interest-rate models and capital asset pricing model (CAPM) are devised [108]. This long string of successes has formed an important bond between the field of engineering and finance. Other famous engineering models have since transcended the boarder between engineering and finance, such as the theory of Fourier transforms which have been adapted to the elegant and efficient pricing of option contracts developed by Carr [13].

In yet other areas, time-series analysis models are finding novel applications. For example, classification of time-series is used in speech recognition to enable functions such as voice activated commands and automated response systems. In modern astronomy, by analyzing the periodicity and spectral content of galactic time-series, astronomers gain a deeper understanding of the underlying mechanisms of various star systems and galactic bodies [29] . Classification and pattern recognition of time series further allow practitioners to build *content-queryable* databases for various sequential data [60].

Although time series analysis is a well-researched and developed discipline, new appli-

cations and challenges continue to emerge everyday, touching on ever more facets of our everyday lives. An example of a more recent development includes a music tagging and search service called Shazam® which operates over the mobile phone network. A user can now hold his or her *smart-phone* against the radio or other audio sources, and the excerpt of music is then analyzed and matched against millions of songs in the Shazam® database. The information about the music, such as song title, singer, and even down to the instance of a particular performance (e.g. live concert performance vs. studio recording), can be returned to the user in a matter of seconds via a text message [106].

In recent years, Gaussian processes (GPs) have made great strides in helping to solve problems in many areas and have gained prevalent acceptance. Researchers of GPs cleverly exploited the closure property of Gaussian random variables, under marginalization and conditioning, for efficient learning and inference. Model selection can be handled naturally under a Bayesian framework, and missing data can be treated effortlessly. The same closure property also limits the GPs to make predictions with Gaussian distributions. However, the world is an inherently non-Gaussian place. The Gaussian prior assumption is simply not valid in many real-world applications, such as the modeling of financial data[69], wind-speeds and temperatures, and frequency and extent of catastrophic damages (e.g. floods, earth-quakes and forest fires[3] [93]). In many cases, it has been shown that risk-management based on Gaussian assumptions severely underestimates the real-risks and greatly inflates the social and economical costs. The ability to accommodate non-Gaussian features is therefore of practical and academic importance.

This weakness has been acknowledged by the machine learning community and many attempts have been made to *upgrade* the *standard* GPs for non-Gaussian situations. Further details will be introduced in Chapter 2. However, due to the inherent Gaussian nature of GPs, most of these extensions are presented as after-thoughts, which are unnatural and computationally inefficient. A principled reformulation is needed.

Econometricians have built up a very respectable arsenal of time-series analysis ma-

chineries of their own as well. The most prominent tool is the Nobel-winning auto-regressive conditional heteroskedastic (ARCH) model by Robert F. Engle [25], which sparked furious waves of applications and extensions, and received well in excess of eight thousand citations. The generalized version of ARCH (GARCH) is now one of the most commonly used models for forecasting volatilities. Nevertheless, there are some shortcomings with the GARCH model. One of these is that the number of parameters scales linearly with the extent of long-range memory, which restricts its practical use in applications where capturing long-range dependencies is desired.

To this end, I propose a principled framework, coined Kernel-based Copula Processes (KCPs), which allows non-Gaussian time-series data to be modeled in a compact manner. The contributions of this thesis are multi-faceted. The KCPs take advantage of the unique properties of copula functions and the power and flexibility of kernel functions to cleanly separate the co-dependent structure of random variables from the marginal distribution while capturing virtually any complex long-range codependency within a compact framework. Further, the KCPs can be extended to analyze a large collection of time-series, as well as make predictions for individual ones. The results have been very encouraging. Not only have the KCPs shown significantly superior predictive powers compared to other state-of-the-art models by different model selection metrics including AIC/BIC, likelihood ratios, and probability integral transforms (PIT), but the KCPs are also able to recover complex dependence structures among multiple time-series. Moreover, the versatile framework of the KCP allows itself to be applied to the classification problem with challenging data sets.

The rest of the document is organized as follows. Before introducing the KCPs model in detail in Chapter 3, Chapter 2 provides a refresher of popular time-series analysis models and standardizes the notations. In Chapter 3 I introduce multi-variate extensions of the KCPs and the associated sampling, learning, and inference procedures. A rarely discussed method of kernel design using stochastic differential equations will also be

presented. Even though time-series forecasting is the primary focus of this work, it will be demonstrated in Chapter 3 that the KCPs can be used for general regression analysis. Furthermore, it can be recast as a classification tool via a simple adaptation. Thus a brief review of the classic literature on the classification problems will also be included. A series of real-life applications are showcased in Chapter 4 with financial time-series being the primary focus. Finally, possible future extensions of this work and conclusions are presented in Chapter 5.

Chapter 2

Background and Related Work

The literature of time-series analysis is tremendously rich and covers many diverse areas, making a comprehensive review impractical for the purposes here. Thus a few seminal works have been carefully selected based on their relevance to the KCPs either by model framework or the needs and requirements of related areas of application. Focus will be mainly placed on econometrics and machine learning. However, this is in no way limits the versatility of the KCPs proposed by this work. Moreover, as it will be shown in Chapter 3, the KCPs can be used a classification model through a simple modification. Therefore, a short review of the classification literature will also be provided for reference although the focus of this work remains in time-series analysis.

2.1 Stochastic Processes and Basic Notations

This section defines the basic notations of stochastic processes and random fields that will be used throughout this document.

A *random variable* y can be defined as a function that assigns a real number, $y(\zeta)$, to each outcome ζ in the sample space Ω of a random experiment. Specifically, a random variable is defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the set of all possible outcomes; $\mathcal{F}$ is a set of events or σ-*algebra*, with each event being a collection of zero or

more outcomes; and $\mathbb{P}$ is the *probability measure* function, mapping the likelihood of an event to a number between 0 and 1.

A *stochastic process* $y(t, \zeta)$ or $y(x, \zeta)$ is a collection of random variables indexed by the scalar *input feature* or *index set* x or $t \in \mathcal{X}$. Specifically, a stochastic process is defined on a sequence of growing σ-algebras, that is, $\mathcal{F}_s \subseteq \mathcal{F}_t$ for $0 \leq s \leq t$. Such sequence of σ-algebras is known as a *filtration*[1]. For example, if $\mathcal{X}$ is the non-negative real axis $\mathbb{R}_+$, then $y(t, \zeta)$ is a *continuous-time* process. On the other hand, if $\mathcal{X}$ is a countable set (e.g. the set of non-negative integers), then $y(t, \zeta)$ is a *discrete-time* process. In the case that $\mathcal{X}$ is the *ordered* set of non-negative integers, it can be viewed as a *time-series*, thus the mixed use of symbols x and t.

When the *input feature* is a vector quantity or multi-dimensional (i.e. $\mathbf{x} \in \mathbb{R}_+^D$, $D > 1$), the stochastic process is known as a *random field*. To simplify the language in this document, the term *stochastic processes* or simply *processes* will be used to denote both stochastic processes and random fields. This generality applies to most models mentioned in this work, including the KCPs, as these models can be used as general regression or prediction models outside of the context of time-series analysis.

Further, $y(t, \zeta = \zeta_0)$ is simply a function of t, for a fixed $\zeta = \zeta_0 \in \Omega$. As such, $y(t, \zeta = \zeta_0)$ is also known as a *sample path*. To simplify notations, ζ is usually dropped in the remainder of this document and a stochastic process is simply denoted by $y(t)$ or y_t.

In general, bold typeface will be used to denote vector quantities. E.g. $\mathbf{y}$ and $\mathbf{x}$ denote a vector of stochastic processes and input features respectively, i.e. $\mathbf{y} = [y_1, y_2, \ldots, y_N]^T$ and $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$. Further, capitalizing variables represents a matrix of vector variables, e.g. $\mathbf{X} = [\mathbf{x}_1^T; \mathbf{x}_2^T; \ldots; \mathbf{x}_D^T]^T$.

Finally, model specific notations will be introduced as needed in the following sections.

[1] Shreve provides a more intuitive explanation of filtrations in [97]: "A filtration tells us the information we will have at future times. More precisely, when we get to time t, we will know for each set in $\mathcal{F}_t$ whether the true ζ lies in that set."

For a more comprehensive treatment of stochastic processes, please consult Papoulis and Pillai [81].

2.2 Machine Learning Models

In this first review section, a number of machine learning models will be introduced. Emphasis will be placed on the standard form of Gaussian processes as it shares an intimate connection with the KCP. It is followed by an extension of Gaussian processes named *warped Gaussian Processes*, designed to accommodate non-Gaussian data through the introduction of a nonlinear warping function. Further, the linear dynamical systems (LDSs) and hidden Markov models (HMMs) are introduced as examples of *continuous-state* and *discrete-state* state-space models, respectively.

2.2.1 The Standard Form of Gaussian Processes

Gaussian Processes (GPs) can be dated to as far back as the 1880s by the Danish astronomer T. N. Thiele. Nevertheless, even now the popularity of GPs continues to grow. The number of published journal papers in this area has increased from the low-tens in the 1990's to above two hundred articles a year in recent time. Variants of GPs have been applied to areas ranging from face-recognition to robotic control [85]. Much of its popularity can be attributed to the closure property of Gaussian distributions, i.e. the marginal and conditional distributions remain in Gaussian form, thus allowing efficient learning and inference. The incorporation of kernel functions in GPs further enhances its flexibility in adapting to different data sets. In fact, Gaussian processes can be considered as the limit of neural networks with infinite hidden units, with the advantage of much simpler inference operation [77].

A Gaussian process is a stochastic process for which any N samples $\{y_1 = y(\mathbf{x}_1), y_2 = y(\mathbf{x}_2), \ldots, y_n = y(\mathbf{x}_N)\}$, taken at arbitrary points in the input space $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$,

form a set of jointly Gaussian random variables for $N \geq 1$ [41] [85]. That is, the probability density function of the set of observations $\mathbf{y}$ is

$$\mathbb{P}\left(y_1, \ldots, y_N | \, \mathbf{x}_1, \ldots, \mathbf{x}_N\right) = \frac{\exp\left(-\frac{1}{2}(\mathbf{y}-\mu)^T \Sigma^{-1}(\mathbf{y}-\mu)\right)}{(2\pi)^{N/2}\,|\Sigma|^{\frac{1}{2}}} \tag{2.1}$$

where μ and Σ are the mean vector and the $N \times N$ covariance matrix respectively; whereas $|\Sigma|$ denotes the determinant of the covariance matrix. In most analyses, one assumes the mean vector μ to be zero for notational clarity (as it can added back easily). Moreover, for machine learning models, the covariance matrix is often modeled by a kernel function to introduce a sense of ordering or distance. Further, noise power is often added on the diagonal of the covariance matrix, such that $\mathbf{y}$ represents a set of noise-corrupted observations. Together, each entry in the covariance matrix is defined as $\Sigma = K + \sigma_n^2 \mathbf{I}$, where K is known as the *Gram* matrix[2]. Each entry of the Gram matrix is an evaluation of the kernel function at the corresponding parts of the input space, i.e. $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Thus, using these new notations, the joint probability density function of the set of observations $\mathbf{y}$ can be rewritten as

$$\mathbb{P}\left(y_1, \ldots, y_N | \, \mathbf{x}_1, \ldots, \mathbf{x}_N\right) = \frac{\exp\left(-\frac{1}{2}\mathbf{y}^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{y}\right)}{(2\pi)^{N/2}\,|K + \sigma_n^2 I|^{\frac{1}{2}}}. \tag{2.2}$$

Finally, it is important to emphasize that, in the context of time-series analysis and most relevant applications in stochastic processes, the input features $\mathbf{x}$ are not random variables. As such, Equation (2.2) represents the unconditional joint-density of the observations $\mathbf{y}$ at the corresponding parts of the input space $\mathbf{x}$, that is $\mathbb{P}_{Y_1(\mathbf{x}_1),\ldots,Y_N(\mathbf{x}_n)}\left(y_1, \ldots, y_N\right)$. Nevertheless, this notation of conditional probability is kept throughout for consistency with the broader machine learning literature, and serves as a reminder that the set of input features $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is fed into the computation of the Gram matrix K.

[2] In general, given a set of vectors $\{v_i\}_{i=1}^N$, the $(i,j)^{th}$ entry of a Gram matrix K is defined as the inner product of the any two vectors from the set, i.e. $K_{ij} =< v_i, v_j >$. The Gram matrix has many different applications in many fields such as quantum chemistry and control theory. However, in the machine learning context and the context of this work, the term Gram matrix is simply used to denote the relationship with kernel functions, i.e. $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

The term *kernel* function stems from the field of integral operators where the functional operator $\mathcal{T}_k$ is defined as

$$\left(\mathcal{T}_k f\right)(\mathbf{x}) = \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') d(\mathbf{x}'), \tag{2.3}$$

which maps functions $f(\cdot)$ into functions $\mathcal{T}_k f(\cdot)$ [94][85]. In the broader kernel machines or kernel methods literature, kernel functions are often defined as a *reproducing kernel* with a nonlinear mapping associated with a reproducing kernel Hilbert space (RKHS) [94]. However, for the context of this work, it is sufficient to consider the kernel function as a mapping of two arguments $\mathbf{x}$ and $\mathbf{x}'$ from the input feature space $\mathcal{X}$ into $\mathbb{R}$, i.e. $k(\mathbf{x}, \mathbf{x}') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

For a kernel function to be a valid covariance function, it must be *positive semidefinite* or PSD. A kernel function is said to be PSD if it gives rise to a PSD Gram matrix[3] for all $N > 1$.

Further, if the kernel function is *stationary*, then it is a function of only $\mathbf{x} - \mathbf{x}'$, i.e. $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$ with a slight abuse of notation. The use of a stationary kernel function results in a stationary GP. In general, kernel functions can take on many different forms to encode different characteristics in the observed data, as will be shown throughout this work.

Learning or model estimation involves learning the values of the *hyperparameters*[4] of the kernel function once it is chosen. This is routinely done by maximizing the marginal log-likelihood

$$\log \mathbb{P}(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2} \log \left| K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I \right| - \frac{1}{2} \mathbf{y}^T \left(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I \right)^{-1} \mathbf{y} - \frac{N}{2} \log 2\pi \tag{2.4}$$

to a local maximum using methods such as conjugate gradient. A maximum a-posterior (MAP) estimate can also be achieved by including a prior over the hyper-parameters.

[3] Recall that an $N \times N$ PSD matrix K is one such that $\mathbf{a}^T K \mathbf{a} \geq 0$ for all vectors $\mathbf{a} \in \mathbb{R}^N$, and equality is only achieved when $\mathbf{a} = 0$. A symmetric matrix is PSD if and only if all of its eigenvalues are strictly positive.

[4] The parameters of the covariance or kernel function are referred to as hyperparameters to emphasize that they are parameters of a non-parametric model.

Alternatively, the hyperparameters can be integrated out using some Monte Carlo methods [78]. Various approximation methods have been developed for large data sets, such as the Nystrom Method and the Bayesian committee machine (BCM) (see Chapter 8 of [85] for details).

Making predictions on unobserved data $\mathbf{y}^*$ at $\mathbf{X}^* = [\mathbf{x}^{*T}_1, \ldots, \mathbf{x}^{*T}_M]^T$ with GPs involves computing the posterior probability, conditioned on observed data $\left\{\mathbf{y}, \mathbf{X} = [\mathbf{x}^T_1, \ldots, \mathbf{x}^T_N]^T\right\}$:

$$\begin{aligned} \mathbf{y}^*|_{\mathbf{y},\mathbf{X}^*,\mathbf{X}} &\sim \mathcal{N}\left(m(\mathbf{y}^*), cov(\mathbf{y}^*)\right), \\ m(\mathbf{y}^*) &\triangleq \mathbb{E}\left[\mathbf{y}^*|\mathbf{y}, \mathbf{X}^*, \mathbf{X}\right] = K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{y}, \\ cov(\mathbf{y}^*) &= K(\mathbf{X}^*, \mathbf{X}^*) - K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}K(\mathbf{X}, \mathbf{X}^*). \end{aligned} \tag{2.5}$$

where $K(\mathbf{X}, \mathbf{X})$ is the $N \times N$ covariance matrix with the $(i, j)^{th}$ entry equal to $k(\mathbf{x}_i, \mathbf{x}_j)$ as usual, while $K(\mathbf{X}^*, \mathbf{X})$ is the $M \times N$ matrix with the $(i, j)^{th}$ entry equal to $k(\mathbf{x}^*_i, \mathbf{x}_j)$. The noise power is assumed to be independent of the input features $\mathbf{x}$ in this often-used formulation (see [40] for GPs with input dependent noise). These notations are further illustrated in Figure 2.1.

Note that the predictive mean function $m(\mathbf{y}^*)$ is a linear combination of the training set targets $\mathbf{y}$, weighted by $K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}$. Thus the predictions made by the GP is a linear combination of the training set targets[5].

[5]This is not to be confused with *linear models* in which the prediction is a linear combination of the input features.

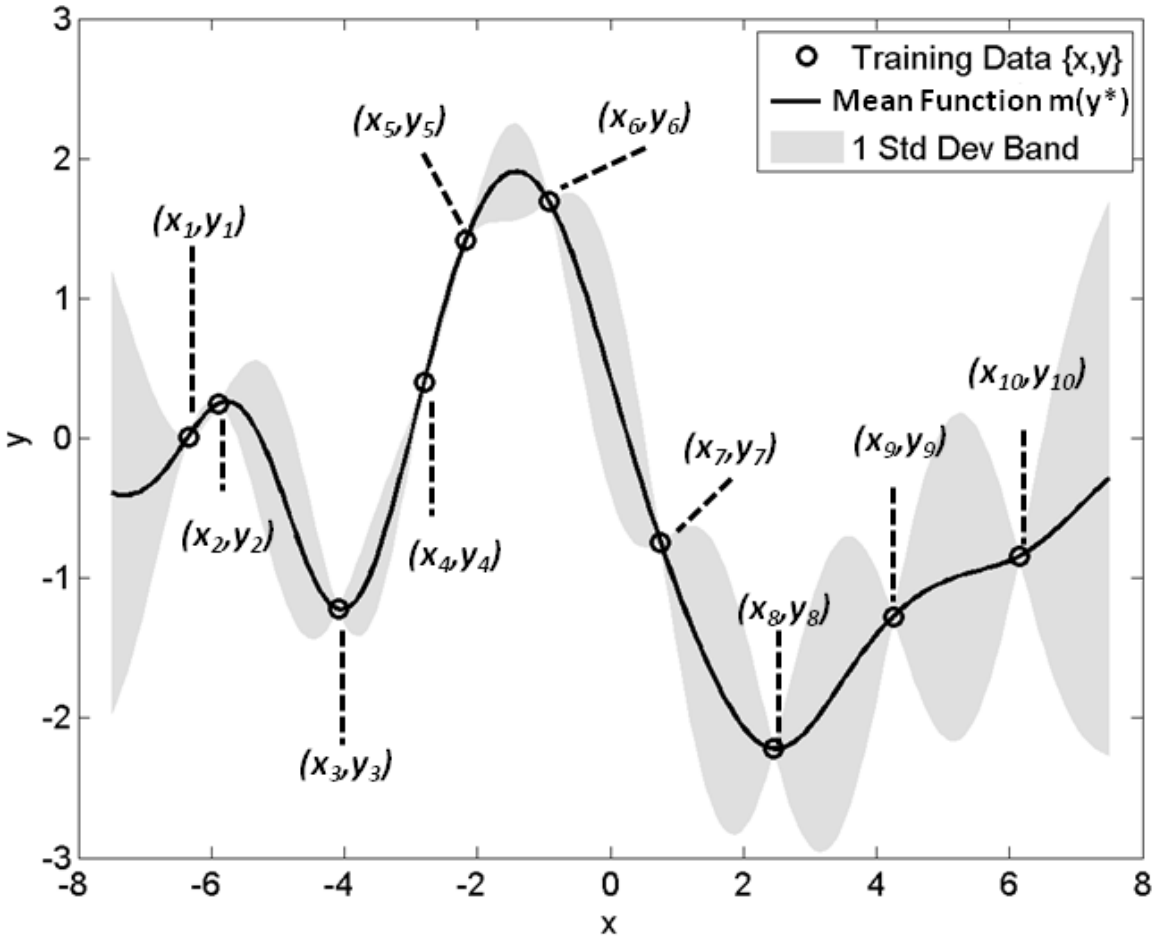

Figure 2.1: Example of a Gaussian process and the associated notations. The training or observed data are denoted as $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{10}$. The solid line denotes the mean function $m(\mathbf{y}^*)$ evaluated at test points $\mathbf{x}^*$ (which is the entire real line in this case). The shaded region is the one standard deviation bounds from the mean function. Note that the standard deviation tightens around the observed data.

2.2.2 Warped Gaussian Processes

Developed by Snelson *et al.* [101], the warped GP addresses the Gaussian limitations of the standard GP by introducing a nonlinear transformation of the GP outputs which in turn allows for non-Gaussian processes and non-Gaussian noise. The central idea here is that the learning algorithm chooses a nonlinear transformation such that the transformed data is well-modeled by a GP. Specifically, consider a vector of latent targets (z) which is modeled by a GP with likelihood function

$$-\log \mathbb{P}(\mathbf{z}|\mathbf{x},\theta) = \frac{1}{2}\log|\Sigma| + \frac{1}{2}\mathbf{z}^T\Sigma^{-1}\mathbf{z} + \frac{N}{2}\log 2\pi, \tag{2.6}$$

where Σ is covariance matrix of the latent at the input $\mathbf{x}$ while $\mathbf{z}$ is the result of the nonlinear transformation h,

$$\mathbf{z} = h(\mathbf{y};\Psi). \tag{2.7}$$

andΨ are the hyperparameters of the transformation. To conserve probability measure in the transformation, the warping function is constrained to monotonic functions such as the logarithmic or the hyperbolic tangent functions. The overall likelihood function becomes

$$-\log \mathbb{P}(\mathbf{y}|\mathbf{x},\theta,\Psi) = \frac{1}{2}\log|\Sigma| + \frac{1}{2}h(\mathbf{y})^T\Sigma^{-1}h(\mathbf{y}) - \sum_{n=1}^{N}\log\left.\frac{\partial h(y)}{\partial y}\right|_{y_n} + \frac{N}{2}\log 2\pi. \tag{2.8}$$

Learning is similar to the standard GPs and can be achieved by simply maximizing the likelihood function in Equation (2.8) using methods such as conjugate gradient. The parameters of the kernel and warping function can be learned simultaneously under the same probabilistic framework. An added requirement, is that the gradient of the warping function is readily computable. Again, the hyperparameters θ and Ψ can be integrated out with Monte Carlo methods just as with the standard GP case.

Making predictions with warped GPs is similar to with the standard GP, but with an additional step. First, prediction is made in the latent variable space:

$$\mathbb{P}(\mathbf{z}^*|\mathbf{z},\theta) = \mathcal{N}\left(m(\mathbf{z}^*), cov(\mathbf{z}^*)\right), \tag{2.9}$$

just as in the case of the standard GP[6]. The distribution is then transformed by the warping function, yielding the final predictive distribution

$$\mathbb{P}(\mathbf{y}^*|\mathbf{y},\theta,\Psi) = \frac{h'(\mathbf{y}^*)}{\sqrt{2\pi cov(\mathbf{z}^*)}} \exp\left[-\frac{1}{2}\left(\frac{(h(\mathbf{y})-m(\mathbf{z}^*))^2}{cov(\mathbf{z}^*)}\right)\right]. \tag{2.10}$$

When a point prediction is required, the result depends on the loss function[7] that is utilized. If the loss function is absolute error, then the median of the prediction distribution is used as the point prediction. If the loss function is squared error, then the mean should be used. For warped GPs, the mean and median can only be obtained easily if the warping is invertible,

$$\mathbf{y}^{mean} = \int h^{-1}(\mathbf{z})\mathcal{N}\left(m(\mathbf{z}^*), cov(\mathbf{z}^*)\right) d\mathbf{z}, \tag{2.11}$$

$$\mathbf{y}^{median} = h^{-1}(m(\mathbf{z}^*)). \tag{2.12}$$

Of course for the standard GP where the predictive distribution is Gaussian, the mean and the median lies at the same point thus yielding the same estimate for both loss functions.

The effectiveness of the warped GP over the standard GP ultimately depends on selecting an appropriate warping function. Snelson *et al.* [101] provided two choices of warping functions:

$$\begin{aligned} h_{nn}(y;\Psi) &= y + \textstyle\sum_{i=1}^{I} a_i \tanh(b_i(y+c_i)), \qquad a_i, b_i \geq 0 \\ h_{splice}(y;\Psi) &= \tfrac{1}{\beta}\log\left[e^{\beta m_1(y-d)} + e^{\beta m_2(y-d)},\right] \quad m_1, m_2 \geq 0 \end{aligned} \tag{2.13}$$

where h_{nn} is the neural-net style function, while h_{splice} is effectively two straight lines of gradients m_1 and m_2, being spliced together at position d with a *curvature* parameter β.

[6]The inputs $\mathbf{x}$ and $\mathbf{x}^*$ are omitted to lessen the notational burden.

[7]This is also known as the (negative) *cost function* or *utility function*. The goal of the decision-making or classification problem is to minimize the total expected loss incurred. The use of a loss function provides a flexible way to tailor loss values according to the severity of different outcomes. For example, if a classifier is applied to cancer diagnosis, a false-negative (i.e. declaring a patient is cancer-free when in fact he/she has cancer) deprives the patient of proper and timely medical care and can be deemed more severe than a false-positive mistake, where the patient may be subjected to further unnecessary tests and anxiety.

However, no clear guideline or logical method has been provided for choosing the warping function that would result in a marginal distribution that matches well with the observations and that would satisfy the invertibility constraint and have computable gradients.

Further, similar to GP, in addition to the limited flexibility in the modeling powers of the marginals, the codependent structure is effectively limited to the Gaussian copula function from the KCP's perspective.

2.2.3 State-Space Models

The origin of state-space models can be traced back to the early 1960's; much earlier than the beginning of the field of *machine learning*. In fact, models such as Kalman filter [57] has its root in engineering and control theory. These models are being included in the machine learning section simply because there have been renewed interests and extensive developments in the machine learning literature in recent years for applications such as speech recognition [51], [64], natural language processing (NLP) [70], and motion analysis [107], as well as new machineries for better learning, such as the expectation maximization (EM) algorithm [92] [38] and the sum-product algorithm [7].

State-space models are generally classified based on the types of latent variables considered. Models with continuous latent variables are known as *linear dynamical systems*[8] (LDSs); whereas models with discrete latent variables are known as the *hidden Markov models* (HMMs). However, both models share the same *graphical model*[9] representation as shown in Figure 2.2.

[8]Kalman filter is probably the best known of all LDS models.

[9]Graphical model representation of probabilistic models is heavily used in machine learning. The graphical model representation not only allows a clear visual representation, it also provides a platform where efficient learning and inference mechanisms can be developed. The particular type of graphical model illustrated in Figure 2.2 is known as the directed acyclic graph (DAG). Each circular node represents a random variable: shaded circles represent observed random variables, while open circles represent latent or unobserved variables. Conditional dependencies are represented by the directed edges, with nodes at the originating end of the arrow designated *parents* and nodes at the terminating end designed

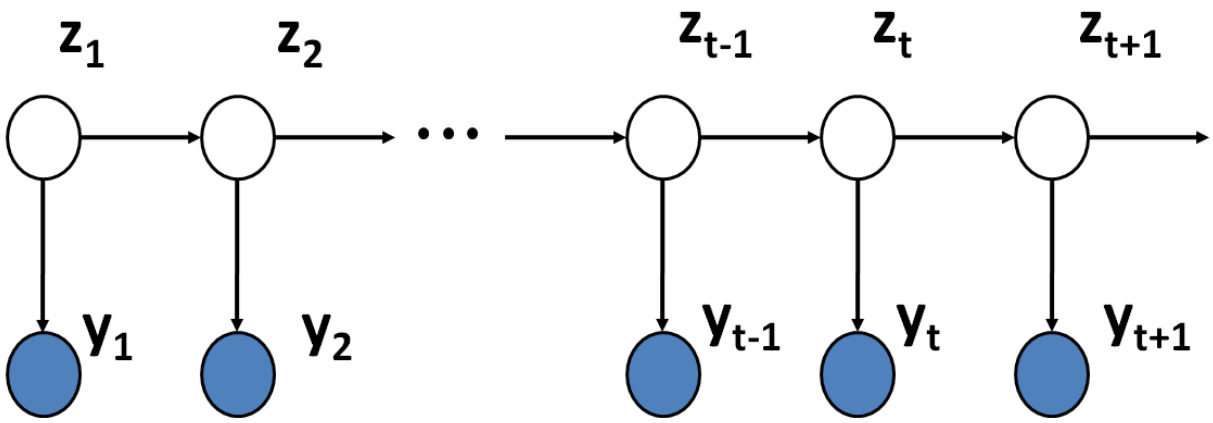

Figure 2.2: The graphical model representation of the state-space model which serves as the common graphical structure for both first-order HMMs and LDSs. $\{\mathbf{z}_t\}$ denotes the latent state process while the $\{\mathbf{y}_t\}$ denotes the observed time-series values.

It is interesting to note that even though the hidden state sequence $\{\mathbf{z}_t\}$ is first order Markov, the observable output process $\{\mathbf{y}_t\}$ is not necessarily Markov of *any* order [91].

The main idea of the state-space models is that the hidden state sequence $\{\mathbf{z}_t\}$ should be an informative lower dimensional projection or explanation of the complicated observation sequence $\{\mathbf{y}_t\}$. With the aid of the dynamical and noise models, the states should summarize the underlying causes of the data much more succinctly than the observations themselves. Please consult Kalman [57], Juang [56], Roweis [91], [92], Bishop [7] for further details.

---

children. For instance, a probabilistic model with a collection of random variables $\{x_1, \ldots, x_n\}$ such that

$$\mathbb{P}(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} \mathbb{P}(x_i | x_{\pi_i}) \tag{2.14}$$

where x_{π_i} is the collection of parent nodes of the node x_i. For a detailed review, please consult [7].

Hidden Markov Models (HMMs)

In addition to viewing the HMMs as a specific instance of the state space model as shown in Figure 2.2 in which the latent variables are discrete, there are two other ways to visualize HMMs as illustrated in Figure 2.3. First, focussing on the hidden state sequence (Figure 2.3a), the HMM can be considered as a Markov chain with stochastic measurements. Whereas when focussing on a single time slot of the chain (i.e. Figure 2.3b), the structure resembles a mixture distribution with the mixture component density given by $\mathbb{P}(\mathbf{y}|\mathbf{z})$, with the added flexibility that the choice of each mixture component at each time slot is not independently drawn but depends on the previous choice of mixture component.

Under this graphical model, the joint distribution factorizes neatly as,

$$\mathbb{P}(\{\mathbf{y}_t\},\{\mathbf{z}_t\}) = \mathbb{P}(\mathbf{z}_0)\prod_{t=0}^{N}\mathbb{P}(\mathbf{z}_{t+1}|\mathbf{z}_t)\prod_{t=0}^{N}\mathbb{P}(\mathbf{y}_t|\mathbf{z}_t). \tag{2.15}$$

The inference operation for HMMs involves taking the observed data sequences as input to produce a probability over the latent state sequence. The problem is substantially more complex than the inference for simple mixture models given the dependencies among the states. In practice, the inference is handled by the forward-backward algorithm, also known as the $\alpha - \beta$ recursion[7].

The parameters of an HMM are the initial probability state distribution $\mathbb{P}(\mathbf{z}_0)$, the parameters for the emission probabilities $\mathbb{P}(\mathbf{y}_t|\mathbf{z}_t)$, and the transition matrix[10], which contains the probabilities $\mathbb{P}(\mathbf{z}_{t+1}|\mathbf{z}_t)$ as elements. The expectation maximization (EM) algorithm is generally used to estimate these parameters for HMMs.

Linear Dynamical Systems (LDSs)

The linear dynamical systems (LDSs) have the same graphical model representation as the HMMs. The major difference is that the state variables are continuous in values

[10]Note that if the transition matrix is time independent, such HMMs are known as *homogeneous* HMMs. The assumption of homogeneity is frequently made for simplicity.

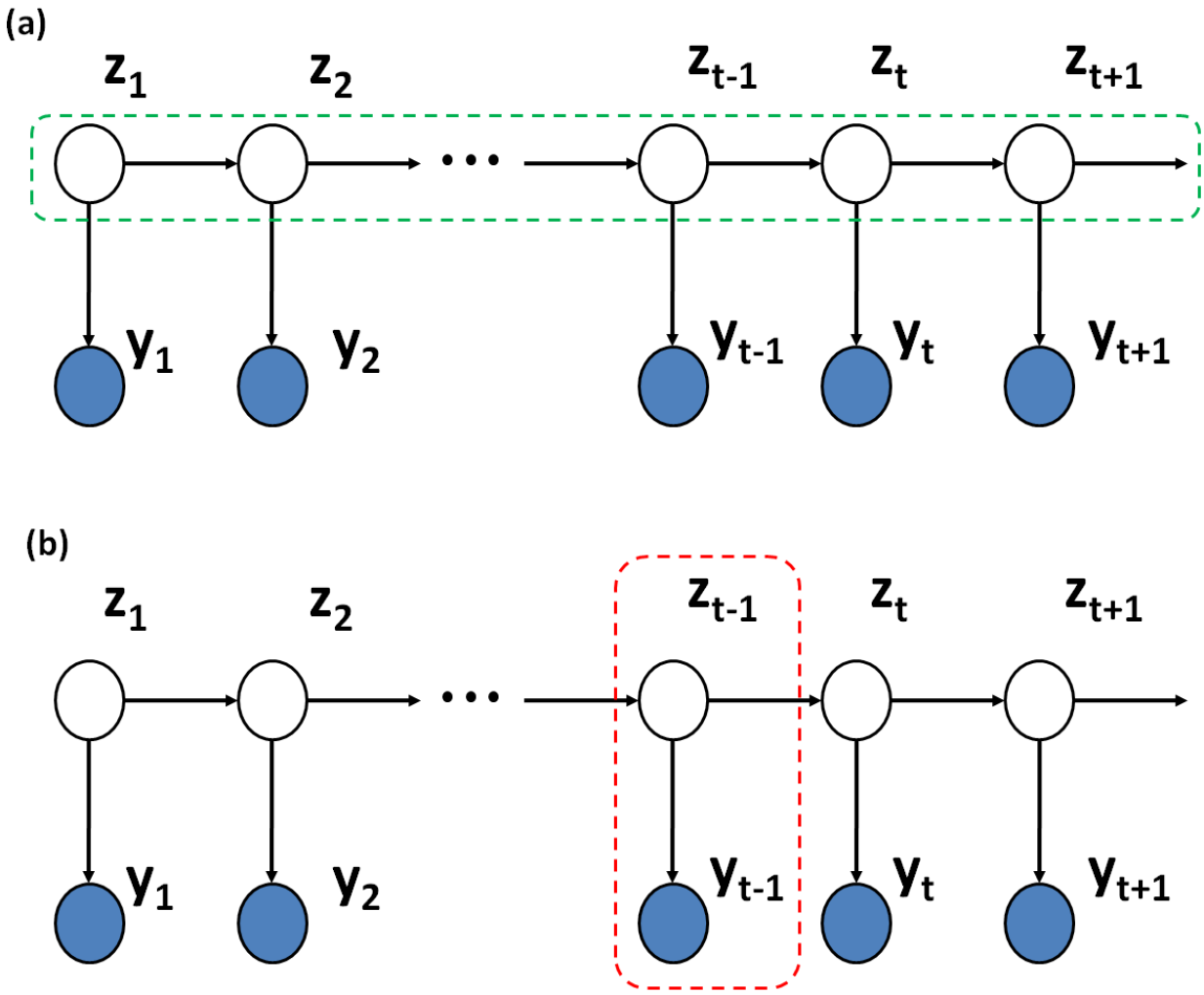

Figure 2.3: Two ways to visualize an HMM: (a) an HMM can be considered as a Markov chain with stochastic measurements; (b) an HMM can be considered as a mixture model where the choice of mixture components depends on the choice from the previous step.

instead of discrete. Further, the observed values are noisy linear transformations of the state variables. The basic generative model can be written as:

$$\begin{aligned} \mathbf{z}_{t+1} &= \mathbf{A}\mathbf{z}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \\ \mathbf{y}_{t+1} &= \mathbf{C}\mathbf{z}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \end{aligned} \tag{2.16}$$

where $\mathbf{A}$ is the $k \times k$ state transition matrix and $\mathbf{C}$ is the $d \times k$ observation measurement or generative matrix. The $\mathbf{w}_t$ and $\mathbf{v}_t$ are the $k \times 1$ and $p \times 1$ state-evolution and observation noises respectively, which are independent of each other and of the values of $\mathbf{z}_t$ and $\mathbf{y}_t$. Both of these noise vectors are essential: without the state noise $\mathbf{w}_t$, the state $\mathbf{z}_t$ would always either shrink exponentially to zero or grow exponentially in the direction of the leading eigenvector of $\mathbf{A}$. Similarly without the observation noise $\mathbf{v}_t$ the state would no longer be hidden. Note that the matrix $\mathbf{Q}$ should be diagonal to avoid degeneracy, otherwise all the structure of the hidden states can be absorbed into the covariance matrix and render the model unintelligent. The matrix $\mathbf{R}$, on the other hand, cannot be constrained in the same way since the equation involves actual observations.

The inference problem in LDS is to determine the hidden state sequence given some observations and a set of fixed model parameters. This operation is known as *filtering* if no observations from the future are included in the information set, and known as *smoothing* otherwise. Filtering and smoothing have been extensively studied for LDSs in the signal processing and engineering communities, starting with the seminal work of Kalman [57], followed by many others such as Rauch [87] [88], Gelb [34], and Juang [56].

The learning problem in LDS, also known as *system identification*, refers to the estimation of model parameters given only the observation sequence. The parameters to be learned are, the matrices $\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}$ and the distribution of the initial state $\mathbf{z}_0$. The traditional control engineering treatment of system identification of the LDSs is given in Juang [56]. More recently, the EM algorithm is often used to maximize the likelihood of the LDSs by the machine learning community. Its application to LDSs Juang [56]. was first introduced by Stoffer [98] and recently extended by Hinton [38].

2.3 Econometric Time Series Models

This section provides a review of a number of time-series analysis models that are fundamentally important to econometrics. In contrast with some previous models, most econometric time models are discrete in nature as data typically becomes available in regular discrete intervals for most economic indicators. For instance, the gross domestic product of the United States is available quarterly (estimate revised monthly). The selections presented here include the two most popular models: the generalized autoregressive conditional heteroskedastic (GARCH) model and vector autoregressive (VAR) model. However, the autoregressive moving average (ARMA) model will be reviewed first due to its fundamental importance and relationship to the GARCH model. For all models described in this section, a more detailed treatment can be found in Hamilton [42] and Lutkepohl [67].

2.3.1 Autoregressive Moving Average (ARMA) Models

The study of the ARMA model alone is not particularly interesting as the variety of behaviors it can model is relatively limited. Nevertheless, it serves as an important foundation in time-series analysis. The stability and causality analysis of ARMA is also reminiscent of the linear time-invariant (LTI) systems in digital signal processing [80].

The ARMA model is in fact the merger of the autoregressive (AR) model and the moving average (MA) model. The AR model is based on the idea that the current value of the time series y_t, can be explained as a function of p past values, $\{y_{t-i}\}_1^p$; whereas the MA model is based on the idea that the current value of the time series is a result of averaging the past random shocks $\{\varepsilon_{t-i}\}_1^p$, or *innovations*, that the time-series experiences in the past q time instances. The innovations are assumed to be *iid*. Typically, a standardized Gaussian distribution is used, but there is no inherent restriction. Finally, an ARMA(p, q) process is simply the sum of the two models, where the current value of the time series

is the results of the weighted average of its past of finite history and innovations. More specifically,

$$y_t = \sum_{j=0}^{q} \theta_j \varepsilon_{t-j} + \sum_{i=1}^{p} \phi_i y_{t-i} + c, \tag{2.17}$$

where $\theta_0 = 1$. Note that the ARMA is limited to the causal case, where only information from the past is available due to the fact that one typically does not enjoy the gift of foresight in economic studies. The ARMA model can be more compactly written with the lag operator L, where $L^i y_t = y_{t-i}$:

$$y_t = \mu + \psi(L)\varepsilon_t, \tag{2.18}$$

where

$$\psi(L) = \frac{\theta(L)}{\phi(L)} = \frac{1 + \theta_1 L + \theta_2 L^2 + \ldots + \theta_q L^q}{1 - \phi_1 L - \phi_2 L^2 - \ldots - \phi_p L^p}, \tag{2.19}$$

$$\sum_{j=0}^{\infty} |\psi^j| < \infty, \tag{2.20}$$

$$\mu = \frac{c}{1 - \phi_1 - \phi_2 - \ldots - \phi_p}. \tag{2.21}$$

To prevent over-parameterization, $\theta(L)$ and $\phi(L)$ must have distinct roots. Finally, for the ARMA(p, q) process to be stable, $\theta(L)$ must not have roots inside the unit circle in the complex plane.

2.3.2 Generalized Autoregressive Conditional Heteroskedastic (GARCH) Models

As noted by many empirical researchers [26] [69], the returns of financial time-series typically do not exhibit constant variance or volatility. Not only does volatility tend to change over time, i.e. is *heteroskedastic*, but periods of high and low volatilities tend to be clustered together, which is known as *volatility clustering*. Clearly, the aforementioned ARMA models, due to the assumption of a constant variance, are incapable of modeling

such behaviors. Models such as the autoregressive conditionally heteroskedastic or ARCH model, first introduced by Engle [25] [11], were developed to model changes in volatility. These models were later extended to the *generalized* ARCH, or GARCH models by Bollerslev [9]. Their applications have been wide-spread, and a long list of extensions have been developed since then. In this section, the standard formulation of GARCH will be reviewed.

In the GARCH model, the observed process and its variance or volatility process σ_t are modeled simultaneously:

$$y_t = \sigma_{t|t-1}\varepsilon_t, \tag{2.22}$$

$$\sigma^2_{t|t-1} = \alpha_0 + \sum_{i=1}^{p} \alpha_i y^2_{t-i} + \sum_{j=1}^{q} \beta_j \sigma^2_{t-j}, \tag{2.23}$$

where $\sigma^2_{t|t-1}$ denotes the conditional variance, conditioned on the past return and variance process. Typically the observed process y_t is the logarithmic returns[12], of a certain financial asset price process.

The unconditional variance exists if and only if

$$\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1. \tag{2.24}$$

If this condition is satisfied, y_t has a constant unconditional variance given by

$$\sigma^2_t = \frac{\alpha_0}{1 - (\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j)}. \tag{2.25}$$

The variance of the process depends on the past squared observed returns as well as the past variance itself, and the next observed return is a random draw from the innovation process scaled by the current variance. Theoretically, there is no restriction on the type of distribution used for the innovation process, but Gaussian white noise is most often

[11] R.F. Engle was awarded the Nobel Memorial Prize in Economic Sciences in 2003 for his work in ARCH. The prize was shared with Clive Granger whose contribution was defining the *Granger casuality* and *cointegration* in time-series.

[12] In addition to the commonly used definition of logarithmic returns $y_t = \ln(\frac{p_t}{p_{t-1}})$, other definitions of returns can also be used, e.g. $y_t = \frac{p_{t-1} - p_t}{p_{t-1}}$.

used. Equation (2.22) also satisfies the empirical observation that the returns of a price process are typically uncorrelated [13].

Note that the ARCH model is recovered when q is set to 0 in Equation (2.23). In his seminal paper, Engle [25] assumed the conditional distribution to be normal so that $\varepsilon_t \sim$ *i.i.d.* $\mathcal{N}(0,1)$. However, even with this seemingly restrictive assumption, the unconditional distribution will generally be non-normal. In particular, it is *leptokurtic*, i.e. it has more mass around zero and in the tails than the Gaussian distribution, thus allowing the model to generate processes with characteristics similar to those of many observed time series. Unfortunately, for many observed series, the ARCH process requires a large order p to capture the dynamics in the conditional variances. Bollersev [9] aimed in the GARCH model to achieve greater simplicity.

The similarity of GARCH and ARMA models for the conditional mean can be seen by defining $v_t \triangleq y_t^2 - \sigma_{t|t-1}^2$, substituting $y_t^2 - v_t$ for $\sigma_{t|t-1}^2$ in Equation (2.23) and rearranging terms to obtain

$$y_t^2 = \alpha_0 + \sum_{i=1}^{p}(\alpha_i + \beta_i)y_{t-i}^2 + \sum_{j=1}^{q}\beta_j v_{t-j} + v_t, \tag{2.26}$$

without loss of generality, it is assumed that $p \geq q$ and $\beta_i = 0$ for $i > q$. Equation (2.26) is formally an ARMA(p,q) model for y_t^2.

Estimation of the GARCH model is typically achieved by conditional MLE. Order selection (i.e. choosing the values of p and q) is often aided by measures such as likelihood ratios, Bayesian informationi criterion (BIC), and Akaike information criterion (AIC).

[13] The *efficient market hypothesis* (EMH) postulates that all publicly available information is efficiently processed by the market and its participants. Thus information is duly incorporated in the relevant tradable asset prices. Thus future price changes cannot be predicted based on the currently available information, causing the future price changes to be uncorrelated with the past price changes.

2.3.3 Vector Autoregressive (VAR) Models

The vector autoregressive (VAR) models are the multiple time-series extensions of the autoregressive processes. A VAR model of order p or the VAR(p) is specified by:

$$\mathbf{y}_t = \boldsymbol{\mu} + \sum_{i=1}^{p} \mathbf{A_i} \mathbf{y}_{t-i} + \boldsymbol{\varepsilon}_t, \tag{2.27}$$

where $\mathbf{y}_t$ is now a $D \times 1$ vector of time-series, $\boldsymbol{\mu}$ is a $D \times 1$ vector of the long-run means, $\mathbb{E}[\mathbf{y}_t]$, $\mathbf{A_i}$ are the $D \times D$ mixing matrices for each lag, and $\boldsymbol{\varepsilon}_t$ is a $D \times 1$ vector of a zero mean, *iid* stationary processes. That is $\mathbb{E}[\boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon'}_s] = 0$ for $s \neq t$ and $\mathbb{E}[\boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon'}_t]$ is diagonal.

The VAR models allow multiple time-series with similar dynamics and properties to be modeled together. For instance, time series such as the future prices of commodities with different maturities and term structure of interest rates are particularly well-suited to the VAR model. The interactions among different variates are modeled by the mixing matrices $\mathbf{A_i}$.

To understand the stability condition of the VAR models, consider the simple VAR(1) model and assume the process has been started in the infinite past,

$$\begin{aligned}
\mathbf{y}_t &= \boldsymbol{\mu} + \mathbf{A}_1 \mathbf{y}_{t-1} + \boldsymbol{\varepsilon}_t & (2.28)\\
&= \lim_{j \to \infty} \left[\left(\mathbf{I}_D + \sum_{i=1}^{j} \mathbf{A}_1^i \right) \boldsymbol{\mu} + \mathbf{A}_1^{j+1} \mathbf{y}_{t-j-1} + \sum_{i=0}^{j} \mathbf{A}_1^i \boldsymbol{\varepsilon}_{t-i} \right] & (2.29)\\
&= \left(\mathbf{I}_D - \mathbf{A}_1 \right)^{-1} \boldsymbol{\mu} + \sum_{i=1}^{\infty} \mathbf{A}_1^i \boldsymbol{\varepsilon}_{t-i}. & (2.30)
\end{aligned}$$

From Equation (2.29) to Equation (2.30), first notice that if all eigenvalues of $\mathbf{A}_1$ have modulus less than 1, the sequence $\mathbf{A}_1^i$ for $i = 0, 1, \ldots$ is absolutely summable. Thus the limit of the last term in Equation (2.29) exists in the mean-square sense. This also means that $\mathbf{I}_D + \sum_{i=1}^{j} \mathbf{A}_1^i$ has the limit of $\left(\mathbf{I}_D - \mathbf{A}_1 \right)^{-1}$. This condition is equivalent to $|\mathbf{I}_D - \mathbf{A}_1 z| \neq 0$ for $|z| \leq 1$.

The same stability analysis can be extended easily to the VAR(p) processes with $p > 1$ with a few augmented notations. In particular, any VAR(p) process with $p > 1$ can be

written as a $(D \times p)$-dimensional VAR(1) process:

$$\mathbf{Y}_t = \mathbf{U} + \widetilde{\mathbf{A}}\mathbf{Y}_{t-1} + \mathbf{E}_t, \tag{2.31}$$

where $\mathbf{Y}_t = [\mathbf{y}_t, \mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-p+1}]^T$, $\mathbf{U} = [\boldsymbol{\mu}^T \mathbf{0}]^T$, $\mathbf{E} = [\boldsymbol{\varepsilon}_t^T \mathbf{0}]^T$, with $\mathbf{0}$ being a $1 \times D(p-1)$ zero vector, and finally

$$\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \ldots & \mathbf{A}_{p-1} & \mathbf{A}_p \\ \mathbf{I}_D & 0 & \ldots & 0 & 0 \\ 0 & \mathbf{I}_D & & 0 & 0 \\ \vdots & & \ddots & \vdots & 0 \\ 0 & 0 & \ldots & \mathbf{I}_D & 0 \end{bmatrix}. \tag{2.32}$$

Similar to other models, the estimation of the parameters for the VAR processes can be achieved by maximum likelihood, and order selection (i.e. choosing the values of p) can be achieved by the measures such as likelihood ratios, Bayesian informationi criterion (BIC), and Akaike information criterion (AIC).

2.4 Dynamic Copula Models

A copula function, in its simplest form, can be defined as the joint distribution of two or more uniformly distributed random variables:

$$\mathbb{P}(U_1 \leq u_1, ..., U_N \leq u_N) = \mathbb{C}(u_1, ..., u_N), \tag{2.33}$$

where $\{U_i\}_{i=1}^N$ are uniformly distributed random variables and $\mathbb{C}(\cdot)$ is the copula function. The joint distribution of random variables with different distributions can then be modeled by first transforming the marginal distribution into an uniform distribution with the corresponding marginal cumulative distribution functions (CDFs):

$$\mathbb{P}(X_1 \leq x_1, ..., X_N \leq x_N) = \mathbb{C}(F_1(x_1), ..., F_N(x_N)), \tag{2.34}$$

where $F_i(x)$ is the corresponding marginal CDF of random variable X_i. The Sklar's Theorem [99] states that there always exists a unique copula function representation of a joint distribution as long as all random variables involved are continuous. Please refer to Appendix A, for a brief review of the theory of Copula.

Equation (2.34) exhibits one of the great benefits of copula functions, namely the separation of specification of the dependency structure and the marginal distributions, thus allowing researchers and practitioners to focus their modeling efforts on either dependency structure or the marginal behavior, as they see fit.

The development of copula functions, in fact, predates the use of the term *copula* by A. Sklar in 1959. Indeed, the study of slight variants can be traced back to the early work of Wassily Hoeffding [48] [49] in which Hoeffding studied bivariate *standardized distributions* and found the basic best-possible bounds inequality for these functions (now known as *Fréchet-Hoeffding* bounds), characterized the distributions, and studied measures of dependence that are *scale-invariant*. Unfortunately, Hoeffding's work was deprived of academic attention by the Second World War. Fréchet [32] independently developed many of the same results in 1951. Later, the work on *probabilistic metric spaces* has Sklar and Schweizer in 1959 [99] to similar functions and finally the name *copula* was born. Similar rediscovery of copulae happened on many occasions in more recent history: including Kimeldorf and Sampson [61] under the name of *uniform representations* in 1975 and Galambos [33] and Deheuvels [20] under the name of *dependence functions* in 1978. However, the field of copula research remained largely dormant until its application in actuarial science and the credit derivatives market in recent years. Even so, the use of copula remained *static* in the sense that the codependent structure among random variables is assumed to be constant over time.

The first effort to extend the use of copula to a dynamic setting, to the best of the author's knowledge, can be traced back to Darsow, Nguyen, and Olsen [19], who related copulae to one-dimensional Markov processes by showing that the Chapman-

Kolmogorov equations for the transition probabilities in a real-valued stochastic process can be expressed in terms of products of bi-variate copulae. As detailed in [79], let $\{X_t | t \in T\}$ be a stochastic process and let C_{st} denote the copula of the random variables X_s and X_t, for each s, t in T. Then the following are equivalent:

1. The conditional distribution function $\mathbb{P}(x, s; y, t)$ satisfies the Chapman-Kolmogorov equations:

$$\mathbb{P}(x, s; y, t) = \int_{-\infty}^{\infty} \mathbb{P}(x, s; z, u)\mathbb{P}(z, u; y, t)dz, \tag{2.35}$$

   for all $s < u < t$ in T and almost all x, y in $\mathbb{R}$;

2. For all $s < u < t$ in T,

$$C_{st} = C_{su} \otimes C_{ut}, \tag{2.36}$$

   where the $\otimes$ operator denotes the *copula product*.

The copula product of bi-variate copulae is defined as:

$$(C_1 \otimes C_2)(u, v) = \int_0^1 \frac{\partial C_1(u, t)}{\partial t} \frac{\partial C_2(t, v)}{\partial t} dt \tag{2.37}$$

It can be shown that the copula product of two bi-variate copulae is also a copula [79].

The work of Darsow, *et al.* [19] provided the first elegant connection between copula functions and stochastic processes. Their model was nevertheless limited to first order Markov. Patton [82] [83] applied the theory of copula to econometric time-series in his doctoral work by formalizing the notion of *conditional copulae*, where all of the information sets that the random variables are conditioned on must be the same. His main contribution, however, was to model the dynamic dependency structure with a copula in which the parameters are time-varying. The constituent univariate time-series are modeled by a mix of AR, ARMA, and GARCH processes. The marginal distributions are subsequently joined by a copula function, such as the Joe-Clayton or Gaussian copulae

with the respective parameters driven by an *evolution* process[83] or ARMA process. Patton's work inspired many other similar pursuits, such as the work of Dias and Embrecht's [21] application to high-frequency foreign exchange series, and was used to detect structural changes in dependency during the introduction of the Euro. Fermanian and Wegkamp [30] further extended the notion of conditional copula to *conditional pseudo copula*. A pseudo-copula satisfies all the properties of a copula (see Section Appendix A) except for the marginalization requirement $C(1, 1, \ldots, u_i, \ldots, 1) = u_i$. This helps to relax the identical information set mentioned above.

Totouom and Armstrong [105] took a radically different approach, using the connection between Laplace transform and the multivariate Archimedean copula discovered by Rogge and Schonbucher [90] to model the default times in a potentially large basket of assets. By modeling a stochastic process that represents the state of an economy, they provided an effective way to model the asymmetric dependency of default times among different companies. Company defaults tend to be highly correlated during a weak economy and uncorrelated when the economy is healthy. This allows better pricing of CDO tranches[14].

2.5 General Comments About Existing Time-Series Analysis Models

Even with a battery of time-series analysis models reviewed here, severe challenges remain to model the real world accurately. For example, all econometric models must be discrete and equally spaced in time. The same is true for HMMs and LDSs. Higher time resolution may solve this problem and make the models suitable for applications such as high-frequency finance, but would dramatically and unnecessarily increase the computational

[14]CDO stands for *Collateralized Debt Obligations* where risks of default of many financial assets (typically corporate bonds or mortgages) are redistributed into different risk levels or *tranches*.

burden. Further, econometric models tends to focus their modeling effect in the dynamics of the second moment, often neglecting higher moments.

On the other hand, while the standard GP is computationally efficient and can handle non-regular time interval data, it cannot satisfactorily cope with non-Gaussian data. The warped GP partially mitigates the problem non-Gaussian nature of real-life data, but provides no principled way of choosing the transformation function or appropriately fitting the model to the data. It is not clear that the warped GP can accommodate any arbitrary data distribution due to the constraints that the warping function must satisfy.

The existing dynamic copula models have made some strides in extending the copula framework to a dynamic setting but have only unleashed a small portion of the modeling power of the copula function. The use of copula was mostly[15] in the cross-sectional sense where copula functions are simply used to join multiple time-series together. The real power of copula, as will be shown in this work, is in the time dimension.

The dynamic copula by Patton *et al.* [82] [83] and its extensions, are driven by an adhoc evolution function through time with foreign exchange rates. No principle method of design was given for that data set or other types of data. Studies have so far focused on bivariate models, and it is not clear how to extend them to a multiple time-series setting. While Patton's model seems to work well with exchange rates with Student's t-copula, it is unclear how it will transfer to other copula families where dramatically different equations may be needed for governing the changing dynamics. Yet another drawback is that the seasonality of the exchange rates must be painstakingly removed [11] and the irregularly spaced high-frequency data must be binned into regular blocks before analysis.

The dynamic copula processes by Totouom and Armstrong [105] have extended Patton's model to the mult-variate case. However, it focussed only on CDO pricing, and

[15]The exception was Darsow's work [19]. Unfortunately, the time dependency was first-order Markov and is thus unable to capture any long-range dependency.

explicitly captures the asymmetry of tightened correlation when default occurs. It is not designed to handle versatile correlation structure in a wide-array of applications.

To this end, I was motivated to devise a novel multiple time-series model that possesses the aforementioned desired features. I named the proposed model kernel-based copula processes (KCPs) for reasons that will become apparent in later chapters. KCPs inherit features from copula functions and GPs to accommodate the stylized facts of different data types (with economic and financial data being the focus in this work), where the distribution could be non-Gaussian with irregular time-spacing. In addition to providing point-estimate forecasts, KCPs are designed to provide the entire predictive distribution. With KCPs, through careful design of kernel functions (see Section 3.3), the preprocessing requirements of data can be minimized to avoid introducing unintended and unwanted distortions. I also provide a logical and principled template for applying the model to different data types. Applications to different real-life data sets, such as temperature time-series and various financial time-series, will also be provided as examples to showcase the versatility of the KCPs.

2.6 Classification

The objective in classification[16] is to assign input feature vectors (continuous or discrete) $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, to one of K discrete classes $\{\mathcal{C}_k\}_{k=1}^K$. It takes two steps to achieve such a goal. First, *training* examples consisting of input vectors $\mathbf{X}$ and corresponding class labels $\mathbf{y} = \{y_i\}_{i=1}^N$ are used to learn the class posterior density $\mathbb{P}(y = \mathcal{C}_k|\mathbf{X})$. Then a classification decision must be made based on the conditional density and some kind of decision criteria, typically known as a *loss* or *utility* function. The separation of the density-modeling from the decision-making allows optimum classification decisions

[16] *Classification* is a form of *supervised learning* where, in addition to input features, class labels and the number of *target classes* are available at the time of learning. Whereas *clustering* is a form of *unsupervised learning* where only the input features are available at the time of learning and the number of target classes or *clusters* are generally not known.

in accordance with the specific application. The specific choice of utility functions and decision criteria given the appropriate probabilities has been the subject of *decision theory*. For more details, please consult Bishop [7] and Berger [6].

The result of classification is that the input feature space is partitioned into regions corresponding to different classes, separated by *decision surfaces*. Thus, the process of training a classifier can be considered as the process needed to define those decision surfaces. Note that the region(s) corresponding to a single class does not, in general, need to be contiguous.

Classification models have a rich taxonomy. First, models can be distinguished by their linearity, that is linear classification models produce linear decision surfaces while nonlinear models are capable of producing more complex decision surfaces. Then, there are frameworks based on the traditional *discriminant functions* versus the more modern probabilistic approach. The discriminant function framework uses a deterministic function to assign input features to different classes. The parameters are often learned using simple optimization techniques such as least square errors. The probabilistic framework, on the other hand, performs classification based on the probability distribution(s) learned from the training data. In fact, the probabilistic classification framework can be further divided into two approaches: *discriminative* and *generative*. The discriminative approach models the class posterior density $\mathbb{P}(y = \mathcal{C}_k|\mathbf{x})$ directly[17]; whereas the generative model first models the joint density $\mathbb{P}(y, \mathbf{x}) = \mathbb{P}(y)\mathbb{P}(\mathbf{x}|y)$, then uses the Bayes' rule to infer $\mathbb{P}(y|\mathbf{x})$. The generative model is so named because one can generate synthetic data by drawing values of $\mathbf{x}$ from the marginal distribution $\mathbb{P}(\mathbf{x})$.

Even though recent research in the machine learning community has focussed on the probabilistic framework, it has provided no definitive answer to settle the debate between the generative and the discriminative approach. This is because both approaches have orthogonal strengths and weaknesses. For instance the discriminative approach is efficient

[17]Hereafter, a simplified notation $\mathbb{P}(\mathcal{C}_k|\mathbf{x})$ will be used to represent $\mathbb{P}(y = \mathcal{C}_k|\mathbf{x})$.

in the sense that it directly models the distribution one needs. This is especially difficult when modeling the class-conditional distribution $\mathbb{P}(\mathbf{x}|y = \mathcal{C}_k)$ with high dimensional input feature $\mathbf{x}$. On the other hand, when dealing with outliers, unlabeled data, and missing input values, it is very helpful to have access to $\mathbb{P}(\mathbf{x})$, which can be obtained by marginalizing out the class labels $\mathbb{P}(\mathbf{x}) = \sum_y \mathbb{P}(y)\mathbb{P}(\mathbf{x}|y)$. The generative approach also allows the incorporation of any prior information that may be available.

The classification approach I developed using the KCPs is discriminative.

In this brief review, attention will be focussed on probabilistic classification. A model from each generative and discriminative approach will be reviewed. The Gaussian Process Classification (GPC) will also be reviewed due to its close relationship to the KCP model. The review given here will focus exclusively on the cases with *continuous input features* $\mathbf{x}$ and are largely based on the treatment in Bishop [7], Jordon [53], Rasmussen and Williams [85].

2.6.1 Generative Classification Model: Mixture of Gaussian

As mentioned above, in the generative approach of classification models, the class-conditional densities $\mathbb{P}(\mathbf{x}|y = \mathcal{C}_k)$ are modeled first. With a choice of class priors $\mathbb{P}(\mathcal{C}_k)$, the posterior probabilities $\mathbb{P}(\mathcal{C}_k|\mathbf{x})$ can then be computed.

First consider the binary classification case where there are only two classes $\{\mathcal{C}_1, \mathcal{C}_2\}$. The posterior probability for class $\mathcal{C}_1$ can be written as:

$$\begin{aligned} \mathbb{P}(\mathcal{C}_1|\mathbf{x}) &= \frac{\mathbb{P}(\mathbf{x}|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(\mathbf{x}|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1) + \mathbb{P}(\mathbf{x}|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)} \\ &= \frac{1}{1 + \exp(-a)} \qquad (2.38) \\ &\triangleq \sigma(a), \qquad (2.39) \end{aligned}$$

where a is defined as $\ln \frac{\mathbb{P}(\mathbf{x}|\mathcal{C}_1)\mathbb{P}(\mathcal{C}_1)}{\mathbb{P}(\mathbf{x}|\mathcal{C}_2)\mathbb{P}(\mathcal{C}_2)}$ and $\sigma(a)$ is the *logistic sigmoid* function. As illustrated in Figure 2.4, the logistic sigmoid function serves as a 'squashing function', mapping the entire real axis into the (0,1) interval as required by probability functions. The

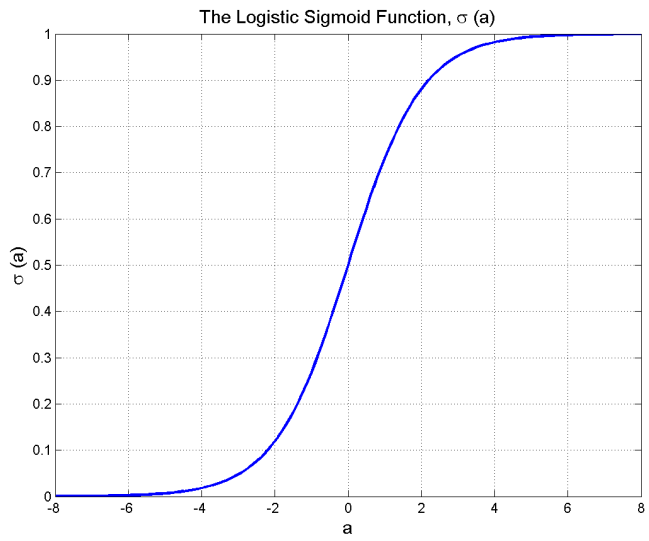

Figure 2.4: The logistic sigmoid function which serves as a 'squashing function' mapping the entire real axis into the (0,1) interval.

inverse of the logistic sigmoid function is known as the *logit* function, and it is given by $\sigma^{-1}(\rho) = \ln\left(\frac{\rho}{1-\rho}\right)$. It represents the logarithmic ratio of the probabilities of the two classes or the *log odds*, $\ln\left(\frac{\mathbb{P}(\mathcal{C}_1|\mathbf{x})}{\mathbb{P}(\mathcal{C}_2|\mathbf{x})}\right)$, making it a function of $\mathbf{x}$, and it is hereafter denoted $a(\mathbf{x}) \triangleq \sigma^{-1}$.

The above analysis is easily extendable to multiple classes:

$$\begin{aligned} \mathbb{P}(\mathcal{C}_k|\mathbf{x}) &= \frac{\mathbb{P}(\mathbf{x}|\mathcal{C}_k)\mathbb{P}(\mathcal{C}_k)}{\sum_{j=1}^{K}\mathbb{P}(\mathbf{x}|\mathcal{C}_j)\mathbb{P}(\mathcal{C}_j)} \\ &= \frac{\exp(a_k)}{\sum_{j=1}^{K}\exp(a_j)}, \end{aligned} \tag{2.40}$$

with a_k defined as

$$a_k = \ln \mathbb{P}(\mathbf{x}|\mathcal{C}_k)\mathbb{P}(\mathcal{C}_k). \tag{2.41}$$

This multi-class generalization is known as *normalized exponential* or *softmax* function.

For the *mixture of Gaussian* model, the central assumption is that the class-conditional density takes on the Gaussian form. That is, if one knows that a data point belongs to a certain class $\mathcal{C}_k$, then the input features follow a Gaussian distribution according to

the parameters associated with that particular class $\theta_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$. Thus the class-conditional density for class $\mathcal{C}_k$ is given by:

$$\mathbb{P}(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}\,|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}, \tag{2.42}$$

where all the classes are assumed to have the same covariance matrix, as will be elaborated shortly.

From Equations (2.41) and (2.42), $a_k(\mathbf{x})$ becomes

$$a_k(\mathbf{x}) = \mathbf{w}_k^T\mathbf{x} + w_{k0}, \tag{2.43}$$

where

$$\mathbf{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k, \tag{2.44}$$

$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln\mathbb{P}(\mathcal{C}_k). \tag{2.45}$$

Note that the quadratic terms in $\mathbf{x}$ from the exponents of the Gaussian densities are canceled out because of the equal covariance matrices assumption across classes. This assumption leads to a linear function of $\mathbf{x}$ in the argument of the softmax function, which ultimately produces linear decision boundaries among classes. If the covariance matrices are allowed to be different for each class, the quadratic terms will remain and the decision boundaries become quadratic accordingly. Both types of decision boundaries resulting from constraints imposed on the covariance matrices are illustrated in Figure 2.5.

The parameters of the mixture of Gaussian models can be learned by maximum likelihood estimation. Assuming the class priors are multi-nominally distributed, i.e.

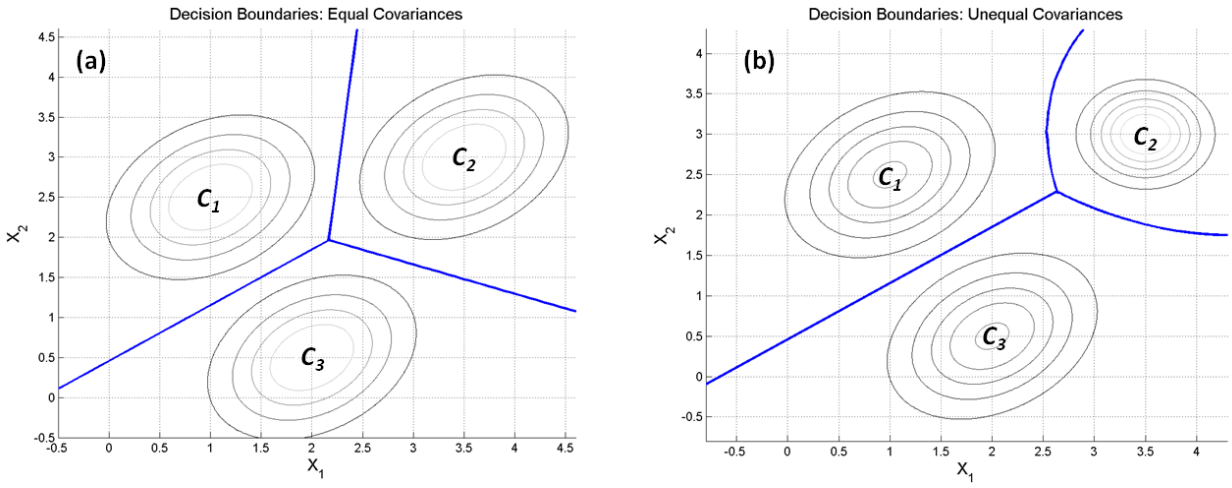

Figure 2.5: Panel (a): Linear decision boundaries as a result of the equal covariance matrices assumption across classes. Panel (b): The decision boundaries become quadratic as this assumption is relaxed.

$\mathbb{P}(\mathcal{C}_k) = \pi_k$ with $\sum_k \pi_k = 1$, then the resulting parameters have the expected forms of

$$\pi_k = \frac{N_k}{N}, \tag{2.46}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \delta_{[y_n, \mathcal{C}_k]} \mathbf{x_n}, \tag{2.47}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \delta_{[y_n, \mathcal{C}_k]} (\mathbf{x_n} - \boldsymbol{\mu}_k)(\mathbf{x_n} - \boldsymbol{\mu}_k)^T, \tag{2.48}$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{\Sigma}_k, \tag{2.49}$$

where N_k and N are the number of training samples belonging to class $\mathcal{C}_k$ and in total, while $\delta_{[y_n, \mathcal{C}_k]}$ is the indicator function which equals to 1 when $y_n = \mathcal{C}_k$ and 0 otherwise. Finally, note that $N_k = \sum_n \delta_{[y_n, \mathcal{C}_k]}$.

2.6.2 Discriminative Classification Model: Logistic Regression

In addition to the more direct approach in modeling the class posterior probability $\mathbb{P}(\mathcal{C}_k|\mathbf{x})$, the discriminative approach may yield improved predictive performance over

the generative approach when there is no good parametric approximation to the true distribution.

In the discriminative approach, no assumption is made regarding the class priors $\mathbb{P}(\mathcal{C}_k)$, class-conditionals $\mathbb{P}(\mathbf{x}|\mathcal{C}_k)$, and marginal probability $\mathbb{P}(\mathbf{x})$. The challenge is to find an efficient way to estimate the class posterior density $\mathbb{P}(\mathcal{C}_k|\mathbf{x})$ directly.

Taking a hint from the generative model discussed in the last section, the logistic regression model assumes a logistic or softmax function at the outset[18]:

$$\mathbb{P}(\mathcal{C}_k|\mathbf{x}, \mathbf{W}) = \frac{e^{\mathbf{w}_k^T \mathbf{x}}}{\sum_{j=1}^{K} e^{\mathbf{w}_j^T \mathbf{x}}}, \tag{2.50}$$

where $\mathbf{w}_k$ is the vector of parameters associated with class $\mathcal{C}_k$, and $\mathbf{W} = [\mathbf{w}_1^T; \mathbf{w}_2^T; \ldots, \mathbf{w}_K^T]^T$. In the generative approach, the above softmax function is a consequence of the Gaussian (or exponential family) class-conditional probabilities assumption. In the discriminative case, no such underlying assumption is made, and the parameter matrix $\mathbf{W}$ is to be estimated directly. To appreciate the properties of logistic regression and its connection to the original regression model, consider the maximum likelihood estimation of the parameters in more details.

To make the notation more compact, rewrite Equation (2.50) as

$$\psi_k = \frac{e^{\eta_k}}{\sum_{j=1}^{K} e^{\eta_j}}, \tag{2.51}$$

where $\psi_k = \mathbb{P}(\mathcal{C}_k|\mathbf{x}, \mathbf{W})$ and $\eta_k = \mathbf{w}_k^T \mathbf{x}$. The derivative of the softmax function in Equation (2.51):

$$\begin{aligned} \frac{\partial \psi_k}{\partial \eta_j} &= \frac{\left(\sum_{m=1}^{K} e^{\eta_m}\right) e^{\eta_k} \delta_{[k,j]} - e^{\eta_k} e^{\eta_j}}{\left(\sum_{m=1}^{K} e^{\eta_m}\right)^2} \\ &= \psi_k \left(\delta_{[k,j]} - \psi_j\right), \end{aligned} \tag{2.52}$$

[18] Just as in the generative approach, there exists choices of class-conditional probabilities that do not yield the softmax form for the posterior density. Other linear functions can be used for the discriminative approach. E.g. the *probit* classification uses the standardized Gaussian CDF or the probit function $\mathbb{P}(\mathcal{C}_k|\mathbf{x}, \mathbf{W}) = \Phi(\mathbf{W}^T \mathbf{x})$.

where $\delta_{[k,j]}$ is equal to 1 if $k = j$ and 0 otherwise. Further, the posterior probability of the k-th class for the n-th data point is given by the multi-nominal probability distribution in the form of:

$$\mathbb{P}(y_n|x_n) = \prod_{k=1}^{K} (\psi_{k,n})^{\delta_{[y_n,\mathcal{C}_k]}}. \tag{2.53}$$

Recall that $\psi_{k,n} = \mathbb{P}(\mathcal{C}_k|\mathbf{x}_n, \mathbf{W})$ and $\delta_{[y_n,\mathcal{C}_k]}$ is the indicator function. Given Equation (2.53), the log-likelihood function can be written as

$$l(\theta|\mathcal{D}) = \sum_n \sum_k \delta_{[y_n,\mathcal{C}_k]} \log \psi_{k,n}. \tag{2.54}$$

Note that in the binary case, the log-likelihood function in Equation (2.54) has the form of a cross-entropy. Now, the gradient of the log-likelihood with respect to the parameter vector $\mathbf{w}_k$ can be calculated with the chain rule:

$$\begin{aligned}
\nabla_{\mathbf{w}_k} l &= \sum_n \sum_m \frac{\partial l}{\partial \psi_{m,n}} \frac{\partial \psi_{m,n}}{\partial \eta_{k,n}} \frac{d\eta_{k,n}}{d\mathbf{w}_k} \\
&= \sum_n \sum_m \frac{\delta_{[y_n,\mathcal{C}_m]}}{\psi_{m,n}} \psi_{m,n}(\delta_{[k,m]} - \psi_{k,n})\mathbf{x}_n \\
&= \sum_n \sum_m \delta_{[y_n,\mathcal{C}_m]}(\delta_{[k,m]} - \psi_{k,n})\mathbf{x}_n \\
&= \sum_n (\delta_{[y_n,k]} - \psi_{k,n})\mathbf{x}_n,
\end{aligned} \tag{2.55}$$

where the fact $\sum_m \delta_{[y_k,\mathcal{C}_m]} = 1$ was used. The final form of the gradient has the same form as the gradient of the sum-of-squares error function for the linear regression model. This result reflects a general property of probability distributions in the exponential family.

It should be mentioned that the standard method for maximizing the log-likelihood in Equation (2.54) is called the *iterative reweighted least squares* which is a form of Newton-Raphson method. More details can be found in Bishop [7] and Jordan [53].

2.6.3 Gaussian Process Classification

The idea behind Gaussian process classification (GPC) is very simple for binary classification. If the class labels $\{\mathcal{C}_0, \mathcal{C}_1\}$ takes on the values $\{0, 1\}$, then one can assume there is

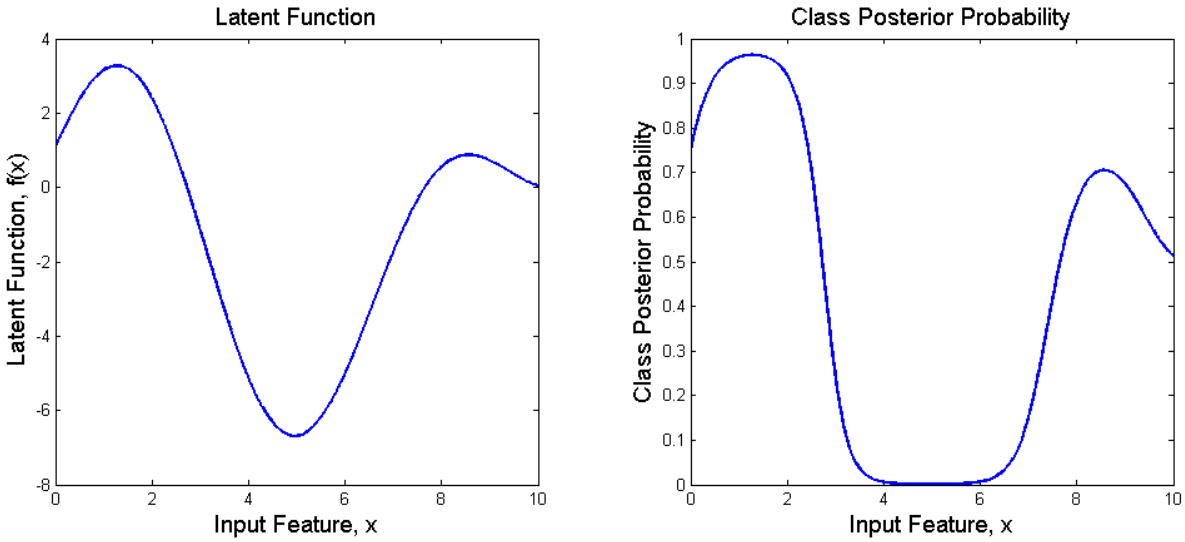

Figure 2.6: Gaussian process classification (GPC): Panel (a) shows a sample latent function $f(x)$ drawn from a GP as a function of x. Panel (b) shows the result after passing it through a squashing function to obtain the class-posterior density.

a *latent function* $f(\mathbf{x})$ which can be modeled by a standard GP. The output of this latent function is then squashed to 0 and 1 by a squashing function such as the logistic function $\sigma(\cdot)$. That is, the class-posterior probability $\mathbb{P}(\mathcal{C}_k|\mathbf{x}) = \sigma(f(\mathbf{x}))$. This relationship is illustrated in Figure 2.6.

Note that one is not interested in nor able to observe the values of the latent function $f(\mathbf{x})$. One, as usual, only observes that training data consisting of the input features and the class labels $\mathcal{D} = \{\mathbf{y}, \mathbf{X}\}$ and cares about the prediction of the class-posterior probability $\mathbb{P}(y^* = \mathcal{C}_k|\mathbf{X}, \mathbf{x}^*)$. The purpose of the latent function is solely to allow for a convenient model formulation, and it will be integrated out later in the process.

Inference is divided into two steps. First the distribution of the latent variable corresponding to a test case in computed:

$$\mathbb{P}(f^*|\mathcal{D}, \mathbf{x}^*) = \int \mathbb{P}(f^*|\mathbf{X}, \mathbf{x}^*, \mathbf{f})\mathbb{P}(\mathbf{f}|\mathbf{X}, \mathbf{y})d\mathbf{f}, \tag{2.56}$$

where $\mathbf{f}$ and f^* are short forms for $f(\mathbf{y})$ and $f(y^*)$, and $\mathbb{P}(\mathbf{f}|\mathbf{X}, \mathbf{y}) = \mathbb{P}(\mathbf{y}|\mathbf{f}, \mathbf{X})\mathbb{P}(\mathbf{f}|\mathbf{X})/\mathbb{P}(\mathbf{y}|\mathbf{X})$

is the posterior over the latent variables. Then this distribution over the latent f^* is used to produce a prediction density

$$\mathbb{P}(y^* = \mathcal{C}_k|\mathcal{D}, \mathbf{x}^*) = \int \sigma(f^*)\mathbb{P}(f^*|\mathbf{X}, \mathbf{x}^*, \mathbf{y})df^*. \tag{2.57}$$

Unlike in the regression case, the predictive distribution remains Gaussian, thus tremendously simplifying the computation. In the GPC model, the non-Gaussian nature of Equation (2.56) and (2.57) makes the calculation analytically intractable. The problem is compounded as the dimension of the input features and the number of classes go up. The computation typically requires some kind of analytic approximations of the integrals or Monte-Carlo based sampling. The two popular approximation methods are *Laplace approximation* by Williams and Barber [110] and *expectation propagation* by Minka [75]. For the Laplace approximation, care must be taken to avoid numerical instability while keeping the computational effort to a reasonable level. The expectation propagation approach is an iterative, yet elegant approach to approximate the class-posterior distribution. Nevertheless it is complex and computationally intensive.

As I will show in Section 3.9, the framework and computational effort of the KCP classification model are virtually identical to the KCP regression model. Thus it is conceptually simple and requires no additional development of heavy computational machineries.

Chapter 3

The KCP Model

This section introduces the main concepts of the kernel-based copula processes or KCPs. To fully appreciate the KCP model, a basic understanding of the theory of copula is essential, as such a brief review is provided in Section 2.4 and Appendix A. Recall that in its simplest form, a copula function is a function that specifies the joint distribution of two or more uniformly distributed random variables. Its importance lies in the fact that it allows the separation of the dependency structure and the marginal distributions in *any* multi-dimensional joint distribution. This separation allows the specification of potentially complex dependencies to be modeled while the particular class of marginal distribution to be chosen independently.

The KCP model surpasses previous works by: first exploiting the unique properties of the copula functions to capture complex and long-range correlations found in real-life random processes; then leveraging the successful framework of Gaussian processes for tackling regression problems while seamlessly accommodating non-Gaussian distributions. As will be revealed in this chapter, the incorporation of kernel functions is the key enabling element of the KCP model.

In addition to the modeling elegance, there are many practical motivations behind the formulation of KCPs. First and foremost, a method for analyzing a collection of

time-series with heavy-tail marginal distribution and complex underlying dependency structure is urgently needed, especially in the area of financial engineering. Further, the learning or estimation of model parameters must be relatively simple and accurate to allow this tool to be used in a high-throughput environment. The nature of copula functions allows the specification and estimation of the dependency structure to be easily separated from those of the marginal distributions.

Section 3.1 provides the formulation for the univariate KCPs, which shows how kernel methods can bring a sense of time or parametric ordering to a high-dimensional copula, thus allowing the modeling of random processes instead of merely a collection of random variables. This approach is reminiscent of the formulation of Gaussian processes (GP) as will be discussed in this section. Indeed, the KCP can be considered as a generalization of the GP. Section 3.2 extends the basic formulation to the multiple time-series case. Furthermore, Section 3.3 presents a novel way of designing new kernel functions from stochastic differential equations. A special sub-class of multivariate KCPs is introduced in Section 3.4. The estimation and inference methods for KCPs are discussed in Section 3.5. We will also address other practical issues such as the handling of missing data and model selection in Sections 3.6 and 3.7. Finally, we take a short detour from time-series analysis to demonstrate the versatility of the KCP framework by applying it to the classification problem in Section 3.9.

3.1 The Univariate KCP

3.1.1 Model Definition

One way to understand the framework of the univariate Kernel-based Copula Process (KCP), is to consider its finite-dimensional distribution: A univariate KCP is a stochastic process for which any finite collections of samples $\mathbf{y} = \{y_1, ..., y_N\}$ taken at

$\mathbf{x} = \{x_1, ..., x_N\}$ form a joint distribution that is defined by a copula function[1] $\mathbb{C}(\cdot)$ and the corresponding marginal distributions $F_i(\cdot)$; that is

$$\boxed{\mathbb{P}(\mathbf{Y} \leq \mathbf{y}|\mathbf{x}) = \mathbb{C}(\{F_i(y_i)\}_{i=1}^N|\mathbf{x}) = \mathbb{C}(\{u_i\}_{i=1}^N|\mathbf{x})} \tag{3.1}$$

where u_i are the transformed y_i via the CDFs, i.e. $u_i \triangleq F_i(y_i)$. Here, the same notation of conditioning on the input features x as introduced in Section 2.2.1 is again used, even though the input features considered in this work are not random variables.

The role of the copula function $\mathbb{C}(\cdot)$ is to capture the entire dependency structure of $\mathbf{y}$. There exists a vast array of parametric or empirical copula functions that allow the specification of complex and heavy-tail dependencies [79]. However, the above finite-dimensional distribution definition alone does not completely define a univariate KCP. The main ingredient that is missing is the ability to describe different types of dynamics through the input feature space, which is precisely the role of the kernel functions in the KCPs. To introduce a sense of ordering/distance (or *time* in the case where the input feature represents time), first the choice of the copula function is restricted to the *elliptical class*, which bears the form of

$$\mathbb{C}(u_1, \ldots, u_n) = G_\Lambda(G_1^{-1}(u_1), \ldots, G_n^{-1}(u_n)) \tag{3.2}$$

where $G_i(\cdot)$ and $G_\Lambda(\cdot)$ are the standardized univariate and zero-mean multivariate distribution function of the same elliptical distribution[2], and Λ is the associated covariance matrix. As a direct consequence of this choice of copula functions, a kernel function can now be embedded in the elliptical copula to specify the covariance matrix and model the dynamics of the process over the input feature space. Specifically, the covariance matrix

[1] Recall that a copula function is a joint distribution function of uniform random variables. Sklar's theorem [79] states that given any joint distribution $H(y_1, \cdots, y_N)$ of random variables $\{y_i\}$ and marginal distributions $\{F_i(y_i)\}$, there exists a copula function $\mathbb{C}$ such that for all y_i in the respective domain, $H(y_1, \cdots, y_N) = \mathbb{C}(F_1(y_1), \cdots, F_N(y_N))$. If $\{F_i(y_i)\}$ are continuous, then $\mathbb{C}$ is unique. The converse is also true.

[2] Please see Section A.5.1 for a definition of elliptical distribution and copula functions.

Λ can be defined as

$$\Lambda = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix} + \sigma_n^2 \mathbf{I} \tag{3.3}$$

where $k(x_i, x_j)$ is a kernel function describing the covariance and σ_n^2 again is the noise power and $\mathbf{I}$ is the identity matrix. The matrix Λ is also known as the *Gram* matrix when its elements are computed using a kernel function. If the noise variance σ_n^2 is greater than zero, the resulting KCP can be regarded as a noise-corrupted observation of some underlying process. However, a very small noise variance σ_n^2 is typically used even when modeling noiseless processes to maintain numerical stability when this covariance matrix is inverted [78]. The use of the kernel function is critical in specifying the degree of similarity of the stochastic process over the input feature space; thus complex behavior such as seasonality and non-stationarity can be described in a compact manner through the careful design of the kernel function. In Chapter 3.3, methods for designing and constructing kernel functions from stochastic differential equations (SDEs) based on the understanding of the dynamics of the data in question will be studied in detail. However, many off-the-shelf kernel functions are also available for general-purpose analysis. Examples of commonly used and versatile kernel functions include the radial basis function (RBF),

$$k_{RBF}(x_i, x_j) = \alpha \cdot \exp\left(-\frac{1}{2h}(x_i - x_j)^2\right), \tag{3.4}$$

and the Ornstein-Uhlenbeck functions,

$$k_{OU}(x_i, x_j) = \alpha \cdot \exp\left(-\frac{1}{h}|x_i - x_j|\right). \tag{3.5}$$

Further examples of kernel functions will be presented throughout this document, but a more comprehensive list can be found in [85].

Before presenting the complete form of univariate KCPs, consider the practical scenarios where the KCPs are typically employed for regression or prediction. In such cases, the definition of the finite-dimensional distribution of the KCPs typically involves

a *training set* $(\mathbf{y}, X) = \{y_i, x_i\}_{i=1}^N$ and a *test set* $(\mathbf{y}^*, X^*) = \{y_i^*, x_i^*\}_{i=1}^N$ with $X \bigcap X^* = \emptyset$, denoting the set of observed and unobserved values and the corresponding input features. The set of observed values is typically used for model learning, and unobserved values are typically the interests of inference and prediction. The operation of learning and inference will be discussed further in Section 3.5. Furthermore, as mentioned earlier in Section 3.1, the input feature space $\mathcal{X}$ of a stochastic process need not be limited to $\mathbb{R}_+$ and can be multi-dimensional (e.g. $\mathbb{R}_+^D$, $D > 1$), in which case it becomes a random field. The same generalization can be afforded by the KCPs. Bold-faced symbols $\mathbf{x}$ and $\mathbf{X} = [\mathbf{x}_1^T; \ldots; \mathbf{x}_N^T]^T$ will be used to represent the individual and the collection of vector-valued input features.

Finally, given the elliptical copula, the Gram matrix (as induced by a kernel function), and the notion of training and test set, the finite dimensional distribution of univariate KCPs in the cumulative distribution function (CDF) form can be written as

$$\mathbb{P}(\mathbf{Y} \leq \mathbf{y}, \mathbf{Y}^* \leq \mathbf{y}^* | \mathbf{X}, \mathbf{X}^*) = \mathbb{C}_\Lambda(\{F_i(y_i)\}_{i=1}^N, \{F_j(y_j^*)\}_{j=1}^M | \mathbf{x}, \mathbf{x}^*) \tag{3.6}$$

$$= \mathbb{C}_\Lambda(\{u_i\}_{i=1}^N, \{u_j^*\}_{j=1}^M | \mathbf{x}, \mathbf{x}^*), \tag{3.7}$$

The joint density of $(\mathbf{y}, \mathbf{y}^*)$ can be obtained by differentiating Equation (3.6) with respect to $\{\mathbf{y}, \mathbf{y}^*\}$:

$$\mathbb{P}(\mathbf{y}, \mathbf{y}^* | \mathbf{X}, \mathbf{X}^*) = \mathfrak{c}_\Lambda(\{F_i(y_i)\}_{i=1}^N, \{F_j(y_j^*)\}_{j=1}^M | \mathbf{x}, \mathbf{x}^*) \prod_{i=1}^N f_i(y_i) \prod_{j=1}^M f_j(y_j^*), \tag{3.8}$$

where $f_i(\cdot)$ are the marginal PDFs and $\mathfrak{c}(\cdot)$ is the so-called *copula density function* given by the derivative of the copula function with respect to its arguments:

$$\mathfrak{c}_\Lambda(\{u_i\}_{i=1}^N) = \frac{\partial^N \mathbb{C}}{\partial u_1 \partial u_2 \ldots \partial u_N}. \tag{3.9}$$

It should be emphasized that the KCP formulation in Equations (3.6) to (3.8), places virtually no constraints or restrictions on the specification of the marginal distribution. Thus, theoretically, the marginal distributions $\{F_j(\cdot)\}$ can be completely flexible and

customized to the particular region of the input feature space. However, in most practical cases, due to the lack of prior information and the preference for model simplicity, the marginal distributions are constrained to be identical, thus reducing $\{F_j(\cdot)\}$ to $F(\cdot)$. A pictorial illustration of the univariate KCP is provided in Figure 3.1 to further clarify the concepts and notations described above.

Finally, the KCPs can be considered as the generalization of the standard form of Gaussian Processes reviewed in Section 2.2.1. In fact, if the Gaussian copula function and Gaussian marginal distributions are used, the KCPs reduce to the GPs exactly. To see this, consider the *Gaussian copula* function, which is given by:

$$\mathbb{C}_\Lambda(\{F_i(y_i)\}_{i=1}^N) = \Phi_\Lambda(\{\Phi^{-1}(F_i(y_i))\}_{i=1}^N) \tag{3.10}$$

where the $\Phi(\cdot)$ and $\Phi_\Lambda(\cdot)$ are the standardized univariate normal CDF and the zero-mean multivariate CDF (with covariance matrix Λ) respectively[3]. Thus if the marginal distributions $F_i(\cdot)$ are chosen to be Gaussian, i.e. setting $F_i(\cdot) = \Phi_{\mu_i,\sigma_i}(\cdot)$ in Equation (3.12), the multivariate joint Gaussian distribution is recovered:

$$\begin{aligned} \mathbb{C}_\Lambda(\{F_i(y_i)\}_{i=1}^N) &= \Phi_\Lambda(\{\Phi^{-1}(\Phi_{\mu_i,\sigma_i}(y_i))\}_{i=1}^N) \\ &= \Phi_{\tilde{\mu},\tilde{\Lambda}}(\{y_i\}_{i=1}^N). \end{aligned} \tag{3.12}$$

where $\tilde{\mu} = [\mu_1, \ldots, \mu_N]^T$ and $\tilde{\Lambda}$ is the full covariance matrix scaled appropriately by the set of marginal variances $\{\sigma_i\}_{i=1}^N$, i.e. $\tilde{\Lambda}_{ij} = \sigma_i \sigma_j \Lambda_{ij}$. Thus the KCPs are theoretically capable of everything GPs are, without the limitation of Gaussian distributions. This is only one of many examples of how the KCPs, through the use of copula theory and the vast and rich selection of elliptical copula and kernel functions, can provide a compact,

[3]As an aside, the corresponding copula density [102] is given by

$$\mathfrak{c}_\Lambda(\{u_i\}_{i=1}^N) = |\Lambda|^{-\frac{1}{2}} \exp\left\{\frac{1}{2}\mathbf{z}^T\left(\mathbf{I} - \Lambda^{-1}\right)\mathbf{z}\right\} \tag{3.11}$$

where $\mathbf{z} = (z_1, \ldots, z_N)$ with $z_i = \Phi^{-1}(u_i)$ for $i = 1, \ldots, N$ and $\mathbf{I}$ is the identity matrix and $|\Lambda|$ denotes the determinant of Λ.

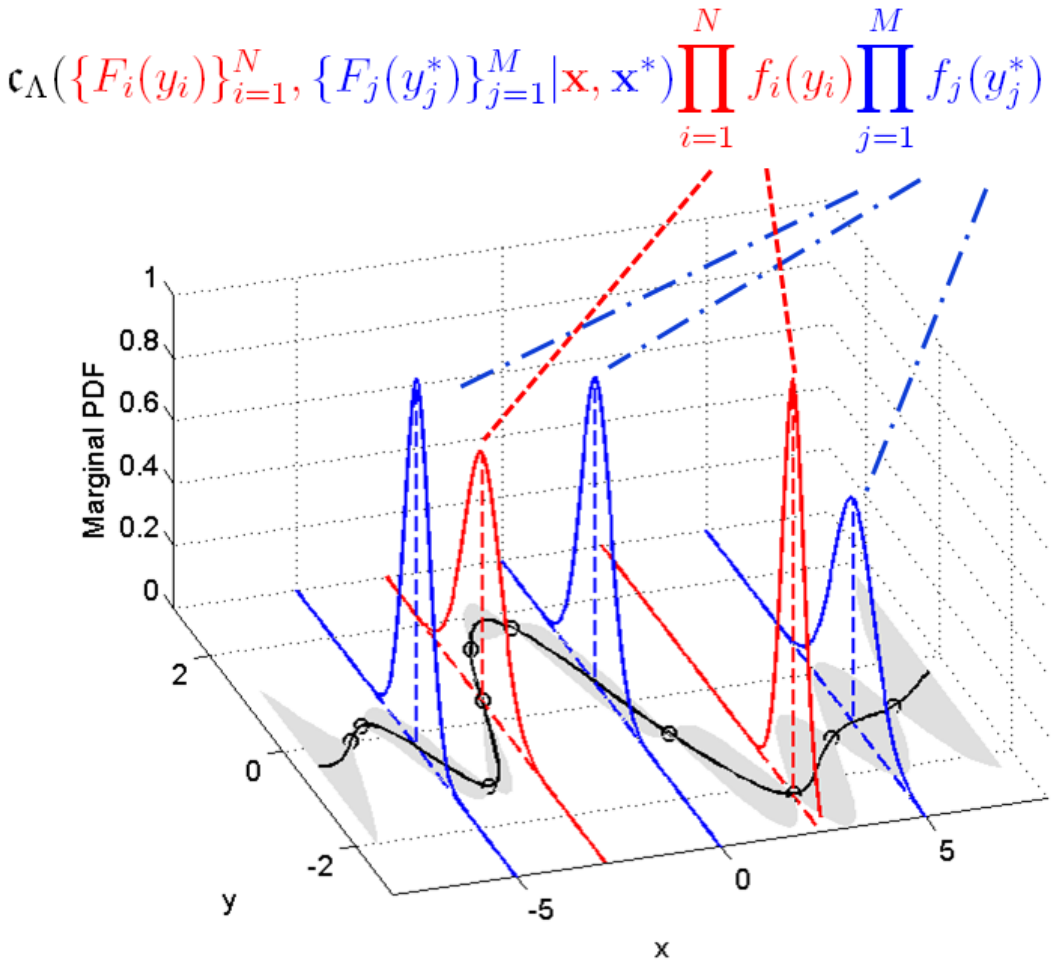

Figure 3.1: A pictorial illustration of a univariate KCP. A prototypical univariate process is drawn on the $x - y$ plane: the observations or training data $\{x_i, y_i\}$ are denoted with circles, while the mean and variance functions as learned by the KCP are denoted by a solid black line and gray-filled area respectively. The constituent parts of a KCP are shown in the vertical dimension: (1) the marginal distribution at each point in the input space x (e.g. time) from which observations are drawn at that particular input; (2) the copula function which joins together the copula functions from across the input feature space.

yet powerful modeling framework. More different instantiations of KCPs and real-life applications will be reviewed throughout this document.

3.1.2 Properties of the Univariate KCP

In this section, the properties of the Univariate KCP will be investigated further in details.

In general, the anatomy of a univariate KCP consists of three parts:

1. the copula function;
2. the kernel function; and
3. the marginal function.

The following sections provide examples and discussions on each of the constituents.

The Role of the Copula Function

The copula function is the heart of the KCP model. It enables the re-parameterization of any joint distribution into a product of its marginal distributions and the co-dependent structure in a compact manner. Further, the KCP model leverages this co-dependent structure to capture any potential long-range dependency.

To illustrate the effects of varying the tail-dependency in the copula function, a set of sample paths are generated using a Student's t-copula with skew-normal marginal, which we denote as TCSM KCP.

The Student's t-copula is given by

$$\mathbb{C}_{\Lambda,\nu_c}(\{F_i(y_i)\}_{i=1}^N) = T_{\Lambda,\nu_c}(\{T_{\nu_c}^{-1}(F_i(y_i))\}_{i=1}^N) \tag{3.13}$$

where the $T_{\nu_c}(\cdot)$ and $T_{\Lambda,\nu_c}(\cdot)$ are the standardized univariate Student's t CDF and the zero-mean Student's t multivariate CDF (with covariance matrix Λ and degree-of-freedom ν_c) respectively.

The skew-normal distribution can be obtained by taking the limit of the skew-t distribution by Hansen[43] as the degree of freedom η approaches infinity. The full formulation of the standardized skew-t distribution is reproduced here for reference:

$$g(z|\eta,\lambda) = \begin{cases} bc\left(1+\frac{1}{\eta-2}\left(\frac{bz+a}{1-\lambda}\right)^2\right)^{-(\eta+1)/2} & z < -\frac{a}{b} \\ bc\left(1+\frac{1}{\eta-2}\left(\frac{bz+a}{1+\lambda}\right)^2\right)^{-(\eta+1)/2} & z \geq -\frac{a}{b} \end{cases} \tag{3.14}$$

where $2 < \eta < \infty$, and $|\lambda| < 1$. The constants a, b, and c are given by $a = 4\lambda c\left(\frac{\eta-2}{\eta-1}\right)$, $b^2 = 1 + 3\lambda^2 - a^2$, and $c = \frac{\Gamma\left(\frac{\eta+1}{2}\right)}{\sqrt{\pi(\eta-2)}\Gamma\left(\frac{\eta}{2}\right)}$.

Each sample path is generated with the same random seed, but using a different value of degree of freedom in the Student's t-copula, ν_c, which in turn controls the extent of tail-dependency. The bottom panels of Figure 3.2, exhibit the enhanced long-range dependency with decreasing ν_c. Not only is the autocorrelation at long lag times stronger, the scatterplot at a long lag time (e.g. 50) also shows a tighter clustering.

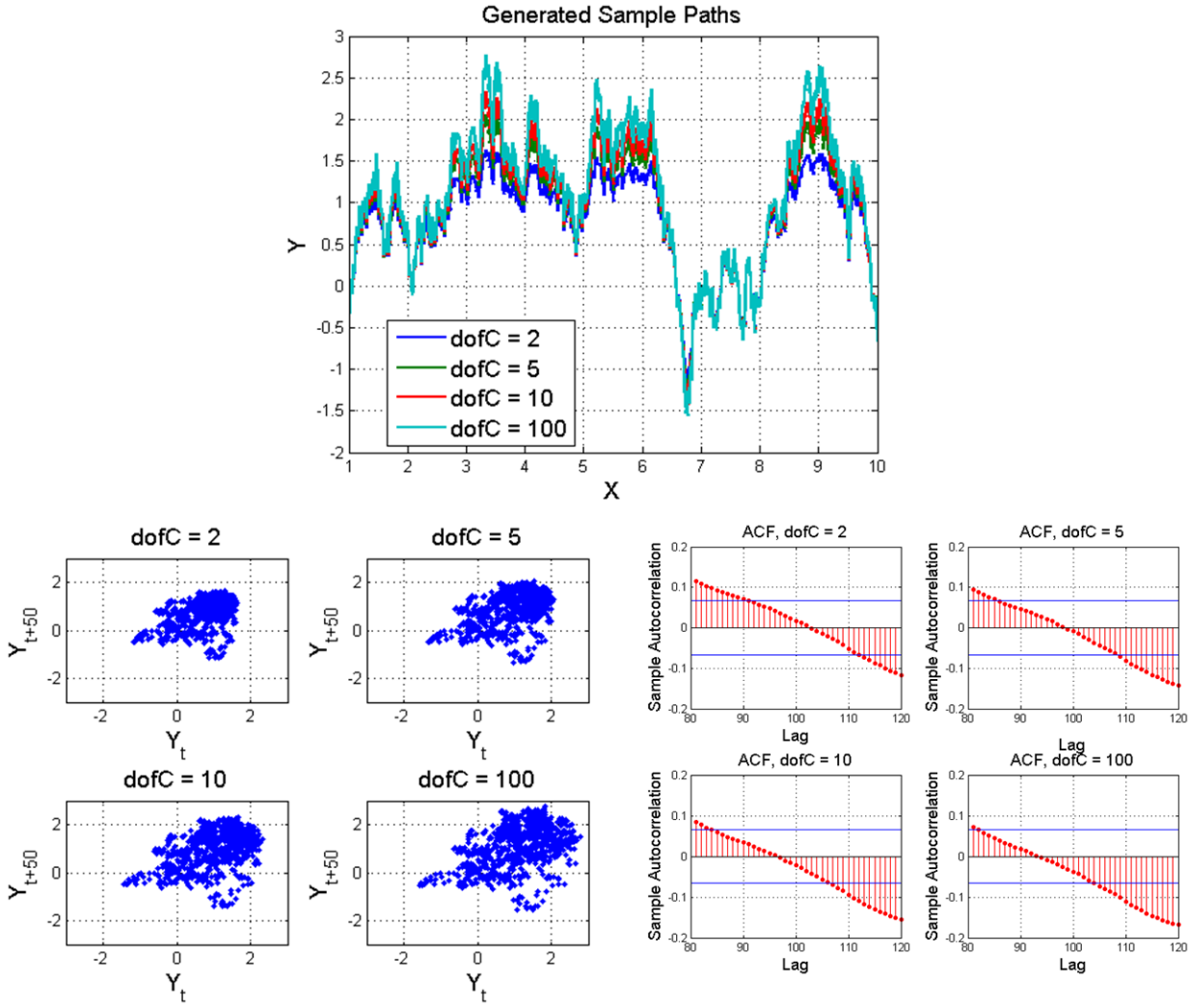

Figure 3.2: Effects of the copula degree of freedom parameter, ν_c. In the top panel, sample paths generated with a TCSM KCP with varying degree of tail-dependency. The tail-dependency is controlled by the degree-of-freedom parameter in the Student's t-copula function. The bottom left panel shows the scatter-plots between the samples at t and $t + 50$ of the same set of sample paths. Notice the clusters become tighter as the tail-dependency is increased. The bottom right panel shows the first order ACFs of the respective sample paths. Only the long-lag portion of the ACFs are shown here. Note that the long-range correlation gets stronger as the tail-dependency is increased.

The Role of the Kernel Function

Recall that in univariate KCPs, the choices for copula function are restricted to the elliptical class (e.g. Gaussian, Student's t, etc.) in order to incorporate the kernel function. The kernel function is a very important pillar of the KCP model for achieving good modeling and predictive performance. As briefly mentioned in Section 3.1.1, the choice of the kernel function is vast and should be tailored to the particular data set being examined. Further details regarding the design of kernel function will be elaborated in Section 3.3. Examples are given in this section to demonstrate the potential influence of kernel functions on KCPs.

First, a few sample paths drawn from the OU kernel (Equation (3.5)) are examined Figure 3.3. Here, three samples with *length-scale* parameter h of values 0.1, 1, and 10 are shown (generated with the same random seed). The larger the value of h, the higher the correlations among distant samples, thus significantly decreasing the volatility or the amount of fluctuations in the sample path. This reaffirms the fact learned from the derivation of the OU kernel that the length-scale parameter is inversely proportional to the *rate of mean-reversion* parameter from the OU process. That is, the higher the mean-reversion rate (the smaller the length-scale parameter), the stronger the autocorrelation and more volatile the sample path appears.

Kernel functions can be used to accommodate different stylized facts[4] in an observed time series such as long-term functional trends and seasonalities. For instance, the following periodic kernel function [85]

$$k(x_1, x_2) = \exp\left(-\frac{A \cdot \sin^2\left(\frac{x_1 - x_2}{T}\right)}{l^2}\right), \tag{3.15}$$

can be used to model the seasonal trends in time-series such as heating oil prices. Some

[4] *Stylized facts* refer to a collection of widely observed empirical facts in data over different markets, asset classes, and time periods. For financial economics, typical stylized facts include leptokurtic returns distributions, clustering of volatilities, and positive volume-volatility correlation.

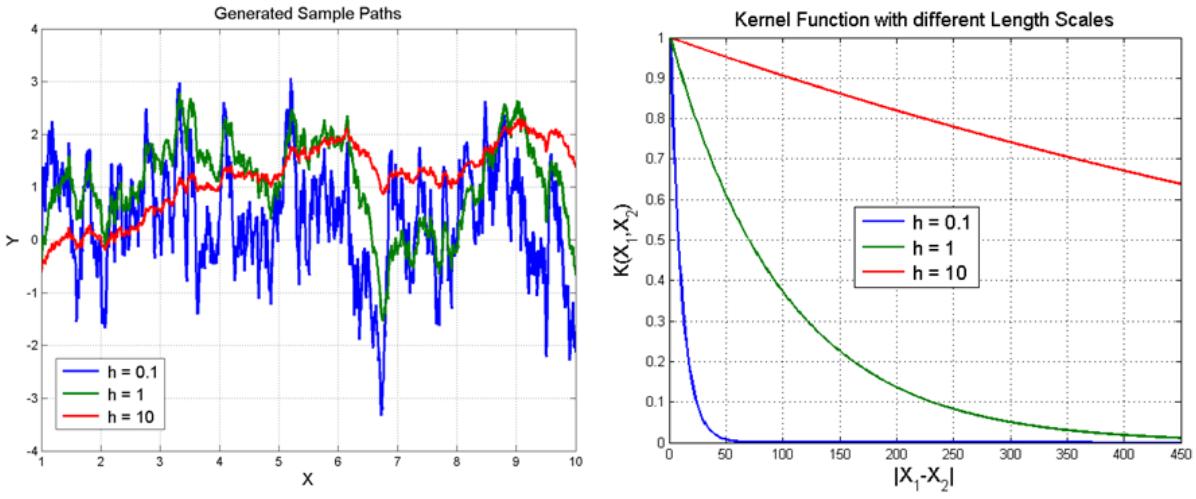

Figure 3.3: Effects of the length-scale parameter h in the OU kernel (Equation (3.5)). The sample paths generated with a TCSM KCP with an OU kernel are shown on the left panel. h takes on values 0.1, 1, 1, while ν_c, λ are fixed at 5 and 0.1 respectively. The corresponding one-sided kernel functions are shown on the right panel.

sample characteristics are depicted in Figure 3.4.

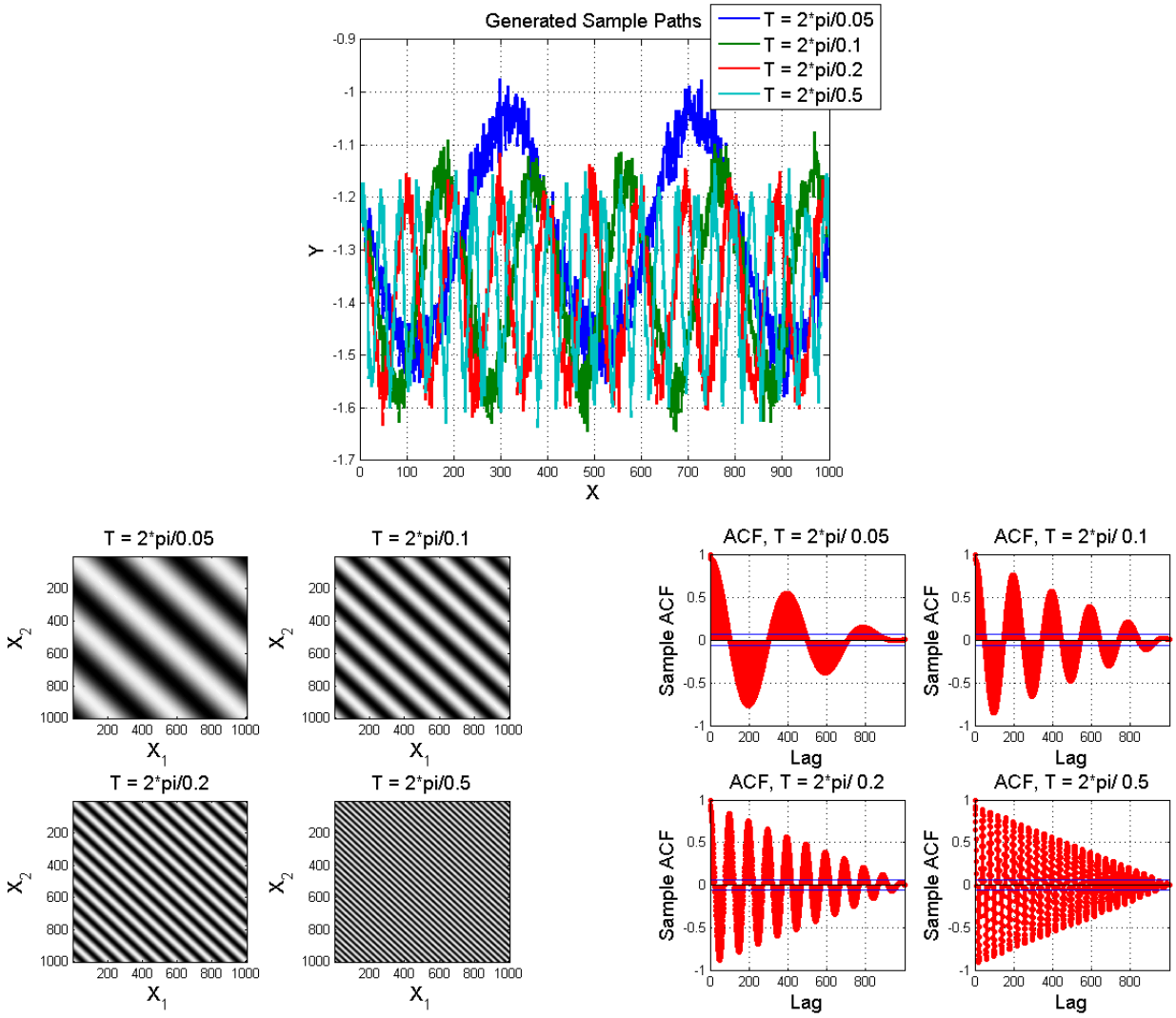

Figure 3.4: Periodic kernel function (Equation (3.15)). The sample paths generated with a GCGM KCP with the period kernel are shown on the upper panel. The period T takes on the values $40\pi, 20\pi, 10\pi, 4\pi$, while A and l are fixed at 2 and 10 respectively. The periodic nature of the resulting processes is manifested at the kernel function image maps and ACFs in the bottom panels.

The Role of the Marginal Function

As its name suggests, the marginal distribution function of the KCP allows one to tailor the marginal behavior of a KCP to match the observed process. This flexibility becomes extremely useful in applications where the marginal distribution of the time-series are clearly non-Gaussian. For example, it is well-known that most financial time-series such as returns of stocks / commodities prices follow some type of heavy-tailed distributions, wind speeds follow the Weibull distributions [23], and electricity prices follow a mixture of heavy-tailed distributions with jumps [109].

The discussion of the role of the marginal distribution function will be deferred to Section 3.1.3, where a synthetic example is constructed to show how the marginal function has direct impact on the shape of the posterior distribution, thereby, adding tremendous flexibility in providing more accurate predictive distributions. Not surprisingly, this added flexibility provides the most compelling reason to deviate from Gaussian distributions and thus the Gaussian processes.

3.1.3 Synthetic Examples

First a synthetic data set is used to illustrate the KCP in action in its simplest form. Here, a sample path is drawn from an AR(2) process with skew-normal (Equation (3.14)) innovations:

$$y_t = 0.1y_{t-1} + 0.05y_{t-2} + 1 + \varepsilon_t \tag{3.16}$$

where ε_t is a standardized skew-normal with skewness of 0.5.

One hundred samples were taken at regular intervals within the range of $t = [-5, 5]$. Every fifth sample is taken as the test set (i.e. the regression / prediction targets), while the remainder of the sample is used as the training set. Then two KCPs: Gaussian copula with Gaussian marginal distribution (GCGM) (or simply a Gaussian process (GP))

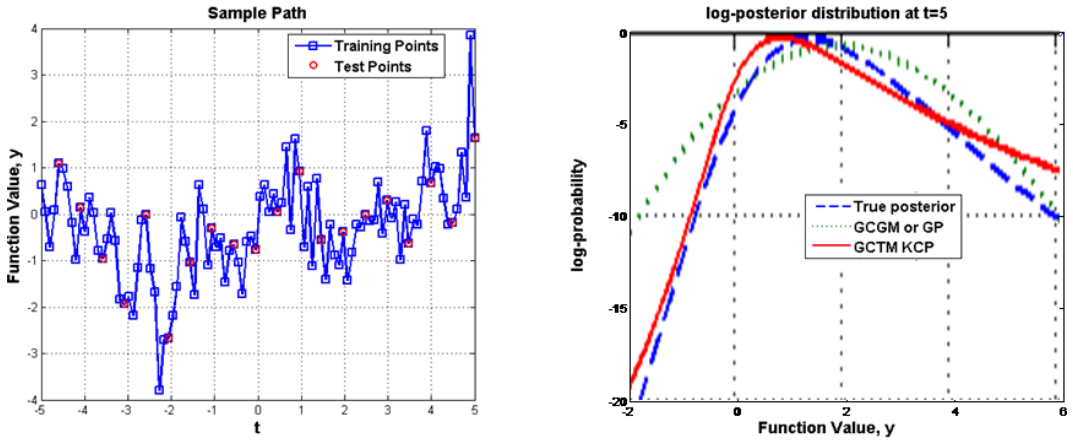

Figure 3.5: A synthetic data set generated by an AR(2) process with skew-normal innovations is shown on the left panel. The posterior distribution of the process at $t = 5$ and the corresponding predictive distributions given by a GP and a GCTM KCP are shown on the right panel.

and Gaussian copula with t-marginal (GCTM), are learnt using MLE. The Ornstein-Uhlenbeck kernel (Equation (3.5)) is used in both cases due to its mean-reverting nature.

To illustrate the superior modeling power of KCPs over GPs, we consider the predictive distribution generated by a GP and a GCTM KCP in Figure 3.5. This experiment is a simple example to illustrate the power and flexibility of copula processes simply by changing the marginal distributions from Gaussian to Student's t. Recall a GP is a special case of the GCTM KCP as the degree of freedom parameter $\nu \to \infty$. The two particular types of KCPs were chosen to illustrate what a slight departure from GPs framework can achieve.

As depicted, the predictive distribution produced by the GCTM KCP is much closer to the true posterior distribution of the AR(2) process than that produced by the GP. The GP, restricted by its symmetric nature, must over shift its mean to the positive side to compensate for the excess probability mass in that region. This would introduce even greater prediction errors if point predictions were considered. The GCTM KCP on the other hand is able to produce an asymmetric predictive distribution based on observed data to much better match the true distribution. Of course, the GCTM KCP is not expected to produce the perfect true posterior distribution as the underlying data generating process (DGP) is quite different. The model is deliberately mis-specified to showcase the power of KCPs even under these harsh conditions. To this end, Section 4.1 will provide more details in selecting the appropriate KCPs for real-life application.

Next, experiments of a larger scale are conducted to explore the modeling power and potential failure modes of the KCPs. In this experiment, 1000 sample paths $\{y_t\}$ with the length of 100 equally spaced samples (i.e. $t = 1, \ldots, 100.$) are generated from each of the four DGPs listed in Table 3.1. For each sample path, a KCP is learned using MLE with the 100 samples. Then the posterior distribution of the true DGPs at $t = 101$ (i.e. $\mathbb{P}(y_{t+1}|\{y_i\}_{i=1}^t)$) is compared with the corresponding predictive distribution of the learned KCPs using the Chi-square test. This is possible because the true DGPs are parametrically known, thus the posterior distribution can be computed exactly. The exercise is then repeated for all 1000 sample paths for each combination of DGPs and KCPs. The percentage of the acceptance of hypothesis that the predictive distribution produced by the KCPs is a good fit for the posterior distribution produced underlying DGPs is recorded is Table 3.1.

In the first two experiments, an AR(2) process and a GARCH (5,5) process with t-innovations are used to emulate the real-life scenario where the real DGP is unknown and inevitably any model choice is incorrect. The same AR(2) process in Equation (3.16)

Table 3.1: List of experiments and the acceptance results at 95% confidence level.

	DGPs	KCPs	% of Acceptance	Median of p-values
1	AR(2) process See Equation (3.16)	TCSM KCP	72.1%	0.032
2	GARCH (5,5), t-innovations See Equation (3.17)	TCSM KCP	21.9%	0.328
3	TCSM KCP $\nu_c = 8, \lambda = -0.5, h = 5, \sigma_n = 0.01$	GCSM KCP	79.4%	0.022
4	TCSM KCP $\nu_c = 8, \lambda = -0.5, h = 5, \sigma_n = 0.01$	TCGM KCP	52.5%	0.0496

is used in Experiment 1, while the GARCH process used in Experiment 2 is given by:

$$\begin{aligned} y_t &= \sigma_t^2 \cdot \varepsilon_t, \qquad (3.17) \\ \sigma_t^2 &= 0.2 + 0.05y_{t-1}^2 + 0.04y_{t-2}^2 + 0.03y_{t-3}^2 + 0.02y_{t-4}^2 + 0.01y_{t-5}^2 + \\ &\quad +0.05\sigma_{t-1}^2 + 0.04\sigma_{t-2}^2 + 0.03\sigma_{t-3}^2 + 0.02\sigma_{t-4}^2 + 0.01\sigma_{t-5}^2 \end{aligned}$$

where ε_t is a standardized Student's t distribution with degree of freedom of 5. The TCSM[5] KCP with an OU kernel is used as the model. In the last two experiments, the objective is to explore the effect of a mismatch between the DGP and the model in both the copula and marginal distribution component of the KCP. As such in Experiment 3, the TCSM KCP, used as the true DGPs, is modeled by the GCSM KCP, thereby introducing a mismatch of copula and hence a mismatch in long-range dependency. This is also a stress-test of the GCSM KCPs as it is less capable of modeling long-range dependency compared to the TCSM KCPs. Similarly, in Experiment 4, the effect of the lack of modeling power in the marginal distribution is explored by using the TCGM KCP

[5]Recall that TCSM KCP refers to a KCP with a Student's t-copula and skew-normal marginal.

to model a TCSM KCP.

First, as preluded earlier the KCP works quite well with AR(2) process. In Experiment 1, a more powerful TCSM KCP (instead of GCTM KCP) is used. As a result, the KCP achieves an acceptance rate of higher than 70% according to the Chi-square test even though there is a clear mismatch of the DGP and model. This high rate of acceptance is partially attributable to the mean-reversion property of the OU kernel. In the second experiment, the acceptance rate is extremely low when modeling the GARCH(5,5) process with the KCP. This failure can be explained by the fact that the GARCH(5,5) process has no autocorrelation in the first moment but very rich dependency in the second moment, while the OU kernel used in the KCP can only model processes with first moment autocorrelation. With other types of kernel functions, however, as will be shown later in this work, the KCPs can be used to model processes with rich correlation structure in the first two moments. However, processes with correlation structure in higher moments but no zero first moment correlation remains a challenge for the KCPs. In such case, the covariance matrix implied by such a process is diagonal, thus reducing the KCP to a model of a simple *i.i.d.* distribution. This reduction causes the copula function KCP to effectively become an independent copula. Thus the predictive distributions it produces are no longer influenced and distorted by the neighboring data points. In the concluding remarks, I will comment further on how the KCPs may be adapted in these scenarios.

Next, in Experiment 3, we used a KCP with reduced long-range correlation modeling power (i.e. GCSM KCP) to model DGPs with higher long-range correlation (i.e. TCSM KCP). The resulting acceptance rate is close to 80%. This high acceptance rate is aligned with the similar exploration performed earlier in Section 3.1.2 (See Figure 3.2) in which that the effects of controlling long-range correlation with the Student's t-copula is relatively subtle. On the other hand, in Experiment 4, an intentional mismatch in marginal distributions seem to have produce a much more dramatic drop in acceptance rate. This

drop in acceptance rate can be mostly attributed to the fact that the DGP has a very high skewness, which makes it very difficult for the Gaussian-bound KCP to adapt.

In some cases, the acceptance rate may have been lowered somewhat by the fact that the MLE learning may not have found the global optimum parameter choice. More random initializations may help in such case.

3.2 The Multivariate KCP

In practical problems, random processes rarely exist in logical isolation but rather typically form an intricate web of dependencies. Thus practitioners are often faced with the challenge of analyzing not a single univariate time-series, but multiple co-dependent time-series. To this end, we propose a multivariate KCP framework to capture such complex interdependencies.

There are many potential choices for modeling the joint probability density in a multivariate data sequence. However, to maintain an elegant connection with the univariate KCP, while simultaneously and naturally incorporating multivariate series, we choose to model the joint conditional distributions for the new set of observations $\mathbf{Y}_t = \{Y_t^{(1)}, \ldots, Y_t^{(D)}\}$ at time t conditional on the past observations $1, \ldots, t-1$. These conditional distributions are then used to build the joint unconditional distribution.

First, we model the joint conditional probabilities using a *binding copula* function as follows:

$$\mathbb{P}\left(\mathbf{Y}_t < \mathbf{y}_t \,\middle|\, \{\mathbf{Y}_i = \mathbf{y}_i\}_{i=1}^{t-1}\right) \triangleq \mathbb{C}_b\left(p_t^{(1)}, \ldots, p_t^{(D)}\right), \tag{3.18}$$

where, $\mathbf{Y}_i < \mathbf{y}_i$ is to be understood component-wise as $\{Y_i^{(1)} < y_i^{(1)}, \ldots, Y_i^{(D)} < y_i^{(D)}\}$,

the function $\mathbb{C}_b(p_t^{(1)}, \ldots, p_t^{(D)})$ denotes the *binding copula* among the sequences and

$$
\begin{aligned}
p_t^{(k)} &= \mathbb{P}\left(Y_t^{(k)} < y_t^{(k)} \;\middle|\; \left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=1}^{t-1}\right) && (3.19)\\
&= \frac{\frac{\partial^{\,t-1}}{\partial y_{t-1}^{(k)} \ldots \partial y_1^{(k)}} \mathbb{C}_k\left(\left\{F_k(y_i^{(k)})\right\}_{i=1}^{t}\right)}{\frac{\partial^{\,t-1}}{\partial y_{t-1}^{(k)} \ldots \partial y_1^{(k)}} \mathbb{C}_k\left(\left\{F_k(y_i^{(k)})\right\}_{i=1}^{t-1}\right)} \\
&= \frac{g_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t}\right) \cdot \cancel{\prod_{i=1}^{t-1} f_k(y_i)}}{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t-1}\right) \cdot \cancel{\prod_{i=1}^{t-1} f_k(y_i)}} \\
&= \frac{g_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t}\right)}{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t-1}\right)} \qquad , k = 1, \ldots, D && (3.20)
\end{aligned}
$$

are the individual (univariate) conditional probability distributions. In the second equality we have written the univariate conditional probabilities in terms of the univariate KCP copula $\mathbb{C}_k$ already learned from the individual sequence $\{Y_i^{(k)}\}_{i=1}^{t}$. Recall that $\mathfrak{c}_k(\cdot)$ is the copula density function of the k^{th} univariate KCP, and $u_i^{(k)} \triangleq F_k(y_i^{(k)})$ while $g_k(\cdot)$ denotes the distribution function resulting from taking the $(t-1)^{th}$ derivatives with respect to $\left\{u_i^{(k)}\right\}_{i=1}^{t-1}$ of the copula function $\mathbb{C}_k\left(\left\{F_k(y_i^{(k)})\right\}_{i=1}^{t}\right)$.

Note that the boundary cases of Equation (3.20) $p_1^{(k)}$ and $p_2^{(k)}$ are defined as:

$$
\begin{aligned}
p_1^{(k)} &= \mathbb{P}\left(Y_1^{(k)} < y_i^{(k)}\right) = F_k(y_1^{(k)}), && (3.21)\\
p_2^{(k)} &= \mathbb{P}\left(Y_2^{(k)} < y_2^{(k)} \;\middle|\; Y_1^{(k)} = y_1^{(k)}\right) = g_k\left(u_2^{(k)}, u_1^{(k)}\right), && (3.22)
\end{aligned}
$$

where in Equation (3.21) we have extended the definition of copula functions with a single parameter to be $\mathbb{C}_k(a) = a, \forall a$, thus yielding $\mathfrak{c}_k(a) = 1, \forall a$. The function $g_k(\cdot)$ may seem numerically intensive at first glance, however this function can take on simple forms especially for elliptical copulae considered under the univariate KCPs. As a concrete

example for $g_k(\cdot)$, consider a univariate KCP with a Gaussian copula,

$$\begin{aligned}
g_k\left(\{u_i\}_{i=1}^t\right) &= \frac{\partial^{\,t-1}}{\partial u_{t-1}\ldots\partial u_1}\mathbb{C}_k\left(\{u_i\}_{i=1}^t\right) \\
&= \left[\frac{\partial^{\,t-1}}{\partial z_{t-1}\ldots\partial z_1}\Phi_{\mathbf{0};\Lambda}\left(\{z_i\}_{i=1}^t\right)\right]\prod_{i=1}^{t-1}\frac{\partial z_i}{\partial u_i} \\
&= \int_{-\infty}^{z_t}\phi_{\mathbf{0};\Lambda}\left(z_1,\ldots,z_{t-1},\alpha\right)d\alpha\cdot\left[\prod_{i=1}^{t-1}\phi_{0;1}(z_i)\right]^{-1} \\
&= \left[\prod_{i=1}^{t-1}\phi_{0;1}(z_i)\right]^{-1}\cdot\int_{-\infty}^{z_t}\frac{\phi_{\mathbf{0};\Lambda}\left(z_1,\ldots,z_{t-1},\alpha\right)}{\phi_{\mathbf{0};\mathbf{A}}\left(z_1,\ldots,z_{t-1}\right)}\phi_{\mathbf{0};\mathbf{A}}\left(z_1,\ldots,z_{t-1}\right)d\alpha \\
&= \frac{\phi_{\mathbf{0};\mathbf{A}}\left(z_1,\ldots,z_{t-1}\right)}{\prod_{i=1}^{t-1}\phi_{0;1}(z_i)}\int_{-\infty}^{z_t}\phi_{\mu;\Sigma}\left(\alpha\right)d\alpha \\
&= \mathfrak{c}_{\mathbf{A}}(u_1,\ldots,u_{t-1})\cdot\Phi_{\mu;\Sigma}\left(z_t\right), \qquad (3.23)
\end{aligned}$$

where $u_i = F(y_i)$ and $z_i = \Phi_{0;1}^{-1}(u_i)$ as usual; the mean vectors $\mathbf{m}$ and covariance matrices $\mathbf{V}$ of Gaussian PDFs and CDFs are explicitly denoted as $\phi_{\mathbf{m};\mathbf{V}}(\cdot)$ and $\Phi_{\mathbf{m};\mathbf{V}}(\cdot)$ to avoid confusion; the Gram matrix Λ of the Gaussian copula is partitioned into

$$\Lambda = \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & b \end{bmatrix}, \qquad (3.24)$$

with $\{\mathbf{A}_{i,j} = k(x_i, x_j)\}_{i,j=1}^{t-1}$, $b = k(x_t, x_t)$, and $\mathbf{C} = [k(x_t, x_1), \ldots, k(x_t, x_{t-1})]^T$. Further, in the last equality of Equation (3.23), using the Gaussian identities for conditional distribution (see Rasmussen and Williams [85] Appendix A.2), it follows that

$$\mu = \frac{z_t}{b}\mathbf{C}, \qquad (3.25)$$

$$\Sigma = \mathbf{A} - \frac{1}{b}\mathbf{C}\mathbf{C}^{\mathbf{T}}, \qquad (3.26)$$

where $\mathbf{z} = [z_1, \ldots, z_{t-1}]^T$. Finally, the superscript (k) on y_i, u_i, z_i are dropped to simplify the notations.

Notice that the conditional probability functions (3.18) and (3.19) are honest probability density functions through the properties of the binding copula function $\mathbb{C}_b(\cdot)$. Furthermore, $p_t^{(k)}$ given by Equation (3.19) is identical to the Probability Integral Transform (PIT) [22], which is often used for model validation as will be described in Section

3.7. Thus the PITs from each of the univariate KCPs serve a dual purpose of (1) model evaluation in the serial sense and (2) binding multiple time-series together in the cross sectional sense.

Additionally, consider the joint density of the current cross-sectional observations $\mathbf{y}_t$ given all of the history $\{\mathbf{y}_i\}_{i=1}^{t-1}$:

$$\mathbb{P}\left(\mathbf{Y}_t = \mathbf{y}_t \,\middle|\, \{\mathbf{Y}_i = \mathbf{y}_i\}_{i=1}^{t-1}\right) = \frac{\partial^{\,D}}{\partial y_t^{(1)} \dots \partial y_t^{(D)}} \mathbb{C}_b\left(p_t^{(1)}, \dots, p_t^{(D)}\right) \tag{3.27}$$

$$= \mathfrak{c}_b\left(p_t^{(1)}, \dots, p_t^{(D)}\right) \cdot \prod_{k=1}^{D} \frac{\partial p_t^{(k)}}{\partial y_t^{(k)}} \tag{3.28}$$

$$= \mathfrak{c}_b\left(p_t^{(1)}, \dots, p_t^{(D)}\right) \cdot \prod_{k=1}^{D} \left[\frac{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t}\right)}{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t-1}\right)} \cdot f_k(y_t^{(k)}) \right]. \tag{3.29}$$

The last equality was obtained by differentiating $g_k(\cdot)$ with respect to $y_t^{(k)}$, finally yielding the product of the copula density $\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t}\right)$ and $f_k(y_t^{(k)})$ through the chain rule. Moreover, following Equations (3.21) and (3.22), we have

$$\mathbb{P}\left(\mathbf{Y}_t = \mathbf{y}_t\right) = \mathfrak{c}_b\left(p_1^{(1)}, \dots, p_1^{(D)}\right) \cdot \prod_{k=1}^{D} f_k(y_1^{(k)}), \tag{3.30}$$

$$\mathbb{P}\left(\mathbf{Y}_2 = \mathbf{y}_2 \,\middle|\, \mathbf{Y}_1 = \mathbf{y}_1\right) = \mathfrak{c}_b\left(p_2^{(1)}, \dots, p_2^{(D)}\right) \cdot \prod_{k=1}^{D} \mathfrak{c}_k\left(u_2^{(k)}, u_1^{(k)}\right) \cdot f_k(y_2^{(k)}) \tag{3.31}$$

Finally, the joint density of the entire collection of observations $\{\mathbf{Y}_i = \mathbf{y}_i\}_{i=1}^{t}$ is given by the telescoping product of conditional densities

$$\mathbb{P}\left(\{\mathbf{Y}_i = \mathbf{y}_i\}_{i=1}^{t}\right) = \mathbb{P}\left(\mathbf{Y}_t = \mathbf{y}_t \,\middle|\, \{\mathbf{Y}_i = \mathbf{y}_i\}_{i=1}^{t-1}\right) \times \mathbb{P}\left(\mathbf{Y}_{t-1} = \mathbf{y}_{t-1} \,\middle|\, \{\mathbf{Y}_i = \mathbf{y}_i\}_{i=1}^{t-2}\right)$$
$$\times \mathbb{P}\left(\mathbf{Y}_{t-2} = \mathbf{y}_{t-2} \,\middle|\, \{\mathbf{Y}_i = \mathbf{y}_i\}_{i=1}^{t-3}\right) \times \cdots \times \mathbb{P}\left(\mathbf{Y}_1 = \mathbf{y}_1\right)$$

resulting in

$$\boxed{\mathbb{P}\left(\{\mathbf{Y}_i = \mathbf{y}_i\}_{i=1}^{t}\right) = \prod_{t'=1}^{t}\left[\mathfrak{c}_b\left(p_{t'}^{(1)}, \ldots, p_{t'}^{(D)}\right) \cdot \prod_{k=1}^{D}\left(\frac{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'}\right)}{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'-1}\right)} \cdot f_k(y_{t'}^{(k)})\right)\right].}$$

(3.32)

Now, let us consider the following pertinent limiting case which helps understand our motivation for the conditional distribution function (3.18) and ultimately the model (3.32). When the binding copula is the independence copula $\mathbb{C}_b(p^{(1)}, \ldots, p^{(D)}) = p^{(1)} \times \cdots \times p^{(D)}$, then the joint distribution function reduces to the product of the univariate KCP distribution functions. To see this, first we note that the copula density of the independence copula is simply:

$$\mathfrak{c}_b(p_t^{(1)}, \ldots, p_t^{(D)}) = \frac{\partial^D}{\partial y_t^{(1)} \ldots \partial y_t^{(D)}} \mathbb{C}_b(p_t^{(1)}, \ldots, p_t^{(D)}) \equiv 1 \tag{3.33}$$

Substituting Equation (3.33) into Equation (3.32) and interchanging the order of the two products yields:

$$\begin{aligned}
\mathbb{P}\left(\{\mathbf{Y}_k = \mathbf{y}_k\}_{k=1}^{t}\right) &= \prod_{k=1}^{D}\left\{\prod_{t'=1}^{t}\frac{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'}\right)}{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'-1}\right)} \cdot f_k(y_{t'}^{(k)})\right\} \\
&= \prod_{k=1}^{D}\frac{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t}\right)}{\cancel{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t-1}\right)}} \times \frac{\cancel{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t-1}\right)}}{\cancel{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t-2}\right)}} \times \cdots \\
&\quad \times \frac{\cancel{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{3}\right)}}{\cancel{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{2}\right)}} \times \cancel{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{2}\right)} \times \prod_{t'=1}^{t} f_k(y_{t'}^{(k)}) \\
&= \prod_{k=1}^{D}\left\{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t}\right)\prod_{t'=1}^{t} f_k(y_{t'}^{(k)})\right\},
\end{aligned} \tag{3.34}$$

which corresponds to simply a product of the univariate KCP models (see Section 3.1).

Conceptually, the multivariate KCP model in Equation (3.32) has a simple and intuitive interpretation. Figure 3.6 provides a schematic depiction of a binding copula joining multiple copula processes together. The advantage of this model is that time-series with greatly different individual dynamics can be captured by the individual copula processes, before they are joined together by the binding copula. For example, one time-series can be the changes in the LIBOR[6] rate while another could be the returns of a stock that is sensitive to interest-rates, such as that of a utilities company or companies with high debt load. In such cases, the stock returns would have much higher volatility but shorter term correlation than the changes in the LIBOR rate. As such, the stock returns and the LIBOR rates can each be modeled by a customized univariate KCP first, then a binding copula can be used to join them together.

[6]The London Interbank Offered rate (LIBOR) is a daily reference rate based on the interest rates at which banks offer to lend to other banks [50].

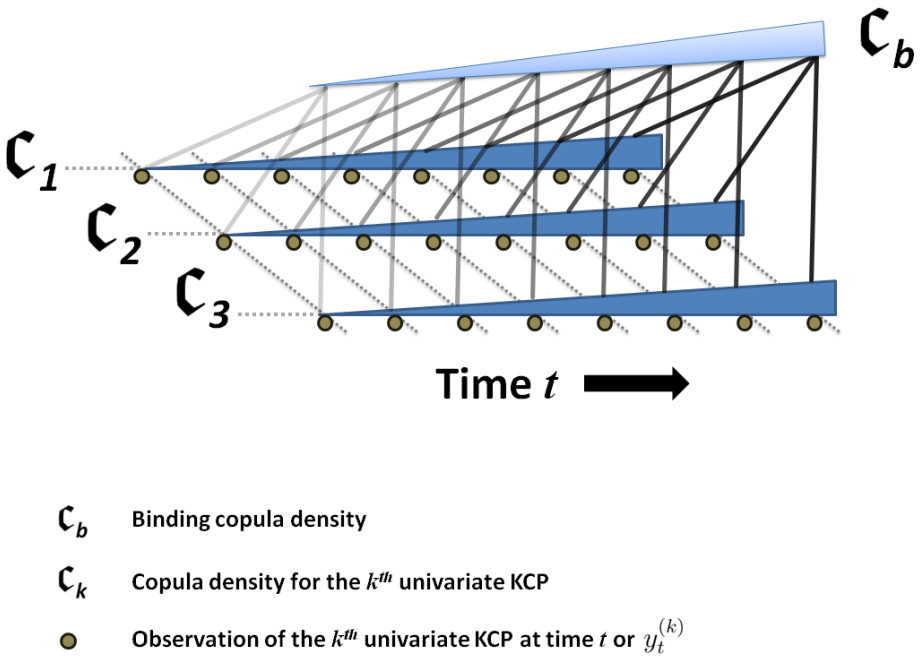

Figure 3.6: A schematic illustration of a multivariate KCP. In this case, three univariate KCPs with elliptical copula functions $\{\mathfrak{c}_1, \mathfrak{c}_2, \mathfrak{c}_3\}$ are joined by a binding copula $\mathfrak{c}_b$. The growing triangles represents the growth of dimensionality of the elliptical copulae as more observations $y_t^{(k)}$ become available. The darkness of the lines connecting the observations to the binding copula presents the growth of information embedded in $p_t^{(k)}$ as more observations are introduced in the conditional probabilities (see Equation (3.18)).

3.3 Kernel Design from SDEs

The kernel function plays the central role in KCP, especially in the univariate case. Rasmussen and Williams [85] have shown how kernel functions can be combined to yield composition kernel functions yielding the desirable properties.

In this section, a different approach is taken – kernels functions are derived from stochastic differential equations (SDEs) having certain desirable dynamics. Two such examples will be provided in detail. One is the powerful and commonly used Ornstein-Uhlenbeck (OU) kernel, and the other one is a non-stationary kernel capable of accommodating periodic changes in variance, which will prove to be very useful in modeling temperature data as showcased in Chapter 4.

3.3.1 Ornstein-Uhlenbeck Kernel

The Ornstein-Uhlenbeck (OU) kernel is also known to be the exponential kernel and a special case of the Matérn class kernel [85]. Not surprisingly, it can be derived from the mean-reverting Ornstein-Uhlenbeck process [63], y_t, satisfying the SDE

$$dy_t = \kappa \cdot (\theta - y_t) \cdot dt + \sigma \cdot dW_t \tag{3.35}$$

where κ is the rate of mean-reversion, θ is the mean-reversion level, σ is the constant variance of y_t, and W_t is a Wiener process in a usual SDE setup. The OU process has the well known solution:

$$y_t = y_0 \cdot e^{-\kappa t} + \theta \cdot (1 - e^{-\kappa t}) + \sigma \int_0^t e^{-\kappa(t-u)} dW_u \tag{3.36}$$

To derive the kernel function, first compute the auto-covariance function by assuming

$t_2 > t_1$:

$$\begin{aligned} Cov[Y_{t_1}, Y_{t_2}] &= \mathbb{E}\left[\left(\sigma \int_0^{t_1} e^{-\kappa(t_1-u)} dW_u\right)\left(\sigma \int_0^{t_2} e^{-\kappa(t_2-u)} dW_u\right)\right] \\ &= \sigma^2 e^{-\kappa(t_1+t_2)} \mathbb{E}\left[\left(\int_0^{t_1} e^{\kappa u} dW_u\right)\left(\int_0^{t_2} e^{\kappa u} dW_u\right)\right] \\ &= \sigma^2 e^{-\kappa(t_1+t_2)} \int_0^{t_1} e^{2\kappa u} du \\ &= \sigma^2 e^{-\kappa(t_1+t_2)} \left(\frac{e^{2\kappa t_1} - 1}{2\kappa}\right) \\ &= \sigma^2 \frac{e^{-\kappa(t_2-t_1)} - e^{-\kappa(t_1+t_2)}}{2\kappa}. \end{aligned} \tag{3.37}$$

Now taking the steady-state limit, i.e. $t_1, t_2 \to 0$ while keeping $|t_2 - t_1| < \infty$), one finds:

$$k(t_1, t_2) = \lim_{\substack{t_1, t_2 \to +\infty, \\ |t_2 - t_1| < \infty}} cov\left(y_{t_1}, y_{t_2}\right) = \frac{\sigma^2}{2\kappa} e^{-\kappa|t_2 - t_1|} \tag{3.38}$$

which is the usual stationary OU kernel.

3.3.2 A Heteroskedastic Kernel

Consider a random process (or time-series) X_t decomposed into a deterministic trend g_t and a stochastic component y_t, i.e. $X_t = g_t + y_t$. Assume y_t follows a common OU mean-reverting process as in Equation (3.35) and has a solution in the form of Equation (3.36), except in this case the variance of the process is a function of time, σ_t. Thus the auto-covariance of X_t at t_1 and t_2 is given by

$$\begin{aligned} Cov[X_{t_1}, X_{t_2}] &= \mathbb{E}\left[\left(\int_0^{t_1} \sigma_u e^{-\kappa(t_1-u)} dW_u\right)\left(\int_0^{t_2} \sigma_u e^{-\kappa(t_2-u)} dW_u\right)\right] \\ &= e^{-\kappa(t_1+t_2)} \int_0^{\min(t_1,t_2)} \sigma_u^2 e^{-2\kappa u} du \end{aligned} \tag{3.39}$$

Notice, that this kernel function is not a function of the difference between t_1 and t_2 and is therefore non-stationary. Now we must specify the form of the variance process. For the detrended temperature data, the variance process is periodic. Consequently, we assume it takes on the following form

$$\sigma_t^2 = \sigma^2 \cdot \left(\alpha \cdot \sin\left(\frac{t}{T} + \phi\right) + 1\right) \tag{3.40}$$

where σ, α, T, and ϕ are all constants. Substituting (3.40) into (3.39), and simplifying gives the final form of the kernel function:

$$\begin{aligned} k(t_1, t_2) &= \sigma^2 \cdot e^{-\kappa \Delta t} \left[\frac{1}{2\kappa} + \frac{\alpha}{4\kappa^2 + T^{-2}} \left(2\kappa \cdot \sin\left(\frac{\min(t_1, t_2)}{T} + \phi \right) \right.\right. \\ &\quad \left.\left. - \frac{1}{T} \cos\left(\frac{\min(t_1, t_2)}{T} + \phi \right) \right) \right] \end{aligned} \tag{3.41}$$

where $\Delta t = |t_2 - t_1|$.

Figure 3.7 provides a few image maps of the heteroskedastic kernel with several periodicities.

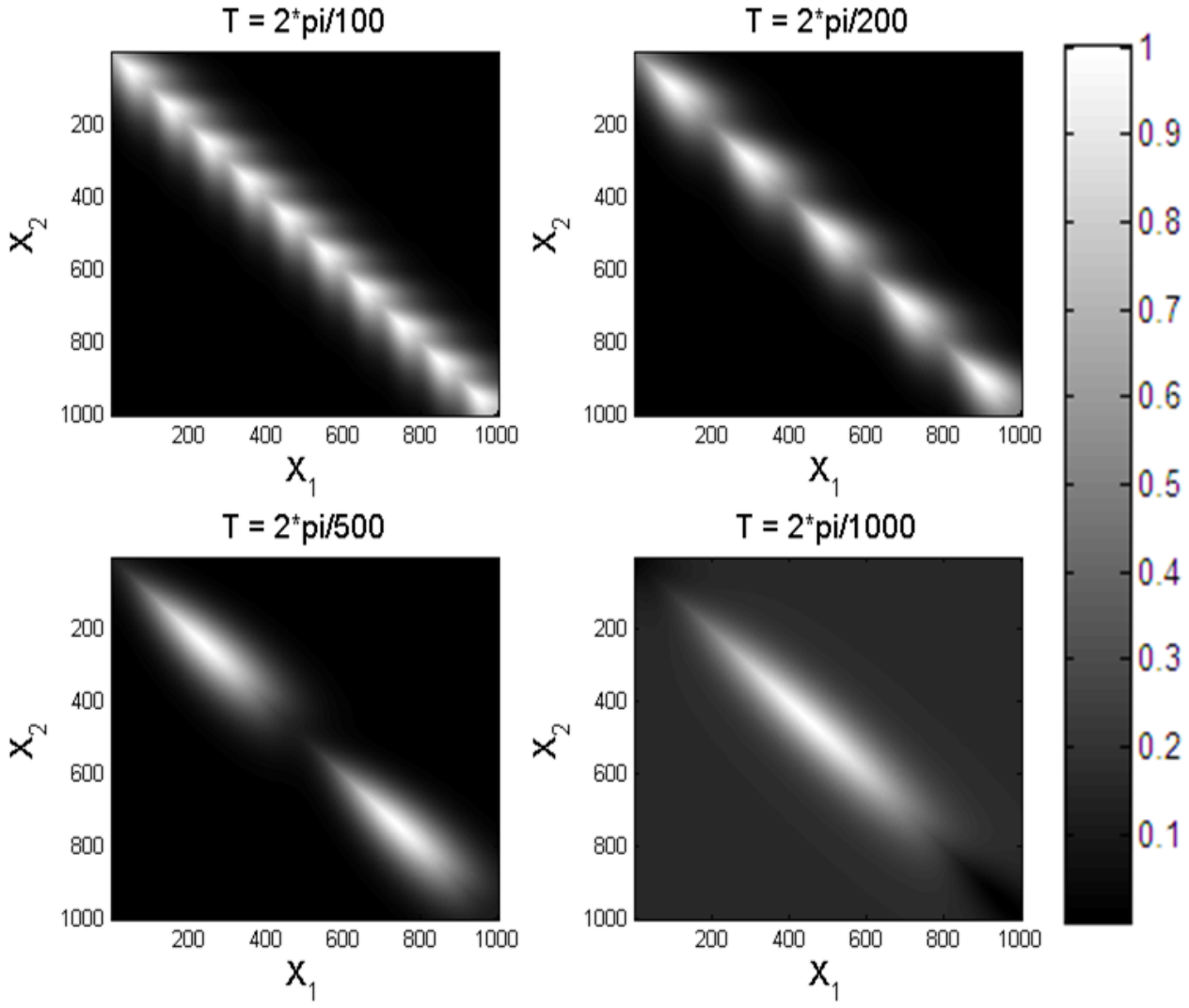

Figure 3.7: Heteroskedastic kernel: the image maps of this non-stationary kernel are shown here with values of T taking one $2\pi/\{100, 200, 500, 1000\}$.

3.4 VAR-KCP

In this section, a subclass of the the KCP model is introduced which makes the connection between KCPs and the popular vector auto-regressive (VAR) model. In fact, the derivation presented below may serve as an illustration of how one may cast any stochastic process that can be represented by an SDE as a KCP.

3.4.1 Model Definition

The vector auto-regressive (VAR) model is defined as follows:

$$\mathbf{X}_t = \mathbf{m} + \mathbf{d}\, t + \mathbf{A}\mathbf{X}_{t-1} + \boldsymbol{\varepsilon}_t \ , \tag{3.42}$$

where $\mathbf{X}_t$ is a $[M \times 1]$ vector of observable time-series under consideration, $\mathbf{m}$ is a $[M \times 1]$ constant mean vector process, $\mathbf{d}$ is a $[M \times 1]$ linear trend vector process, $\mathbf{A}$ is a $[M \times M]$ cross-series interaction matrix, and $\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \boldsymbol{\Omega})$ is the *iid* M-dimensional zero mean normal innovation with $\boldsymbol{\Omega}$ being diagonal.

This system of equations has a continuous-time representation as follows.

$$d\mathbf{X}_t = -\boldsymbol{K}\mathbf{X}_t dt + \boldsymbol{\Lambda} d\mathbf{W}_t, \tag{3.43}$$

where $\mathbf{K}$ is a $[M \times M]$ rate of mean-reversion matrix, $\mathbf{W}_t$ is a $[M \times 1]$ vector of Wiener processes, while $\boldsymbol{\Lambda}$ is a $M \times M$ covariance matrix, which is constrained to be diagonal. Now we derive the solution of this SDE. Assume $\boldsymbol{K}$ can be diagonalized into $\mathbf{U}^{-1}\mathbf{S}\mathbf{U}$ and let $\mathbf{Y}_t = \mathbf{U}\mathbf{X}_t$, then Equation (3.43) becomes

$$d\mathbf{Y}_t = -\mathbf{S}\mathbf{Y}_t dt + (\mathbf{U}\boldsymbol{\Lambda}) d\mathbf{W}_t. \tag{3.44}$$

The system now decouples into one dimensional SDEs:

$$dy_t^{(i)} = -s_i y_t^{(i)} dt + \sum_j (\mathbf{U}\boldsymbol{\Lambda})_{i,j} dw_t^{(i)}, \tag{3.45}$$

where $y_t^{(i)}$ and $w_t^{(i)}$ are the i^{th} processes in $\mathbf{Y}_t$ and $\mathbf{W}_t$ and s_i is the i^{th} entry (or eigenvalue) on the diagonal of $\mathbf{S}$.

Inserting the exponential form,

$$y_t^{(i)} = e^{-s_i t} q_t^{(i)}, \tag{3.46}$$

upon differentiating yields,

$$dy_t^{(i)} = -s_i y_t^{(i)} dt + e^{-s_i t} dq_t^{(i)}, \tag{3.47}$$

where $dq_t^{(i)}$ is given by

$$\begin{aligned} dq_t^{(i)} &= \sum_j (\mathbf{U}\mathbf{\Lambda})_{i,j} e^{s_i t} dw_t^{(i)}, & (3.48)\\ \Rightarrow q_t^{(i)} - q_0^{(i)} &= \sum_j (\mathbf{U}\mathbf{\Lambda})_{i,j} \int_0^t e^{s_i v} dw_v^{(j)}, & (3.49)\\ \therefore y_t^{(i)} &= e^{-s_i t} y_0^{(i)} + \sum_l (\mathbf{U}\mathbf{\Lambda})_{i,l} \int_0^t e^{-s_i (t-v)} dw_v^{(l)}. & (3.50) \end{aligned}$$

Recall that $\mathbf{X}_t = \mathbf{U}^{-1}\mathbf{Y}_t$, so that

$$\begin{aligned} x_t^{(j)} &= \sum_k (\mathbf{U}^{-1})_{j,k} y_t^{(k)} \\ &= \sum_k (\mathbf{U}^{-1})_{j,k} e^{-s_k t} y_0^{(k)} + \int_0^t \sum_{l,k} (\mathbf{U}^{-1})_{j,k} (\mathbf{U}\mathbf{\Lambda})_{k,l} e^{-s_k (t-v)} dw_v^{(l)}. \end{aligned} \tag{3.51}$$

Now, to derive the corresponding kernel function for the KCP, we compute the covariance of time series i and j at t_1 and t_2, i.e. $cov\left[x_{t_1}^{(j)}, x_{t_2}^{(i)}\right]$. Here, we assume the processes have existed sufficiently long and have reached steady-state (i.e. $t_1, t_2 \gg 1$), while the difference between the two times remains finite (i.e. $|t_1 - t_2| < \infty$). Clearly,

$$cov\left[x_{t_1}^{(j)}, x_{t_2}^{(i)}\right] = cov\left[\int_0^{t_1} f(i,k) e^{-s_k (t-v)} dw_v^{(l)}, \int_0^{t_2} f(j,m) e^{-s_m (t-v)} dw_v^{(l)}\right], \tag{3.52}$$

where $f(i,k) = \sum_{l,k} (\mathbf{U}^{-1})_{i,k} (\mathbf{U}\mathbf{\Lambda})_{k,l}$. Let $\tau = \min(t_1, t_2)$. Then by Itô's isometry [7], we

[7] Itô's isometry states that

$$\mathbb{E}\left[\left(\int_0^T X_t dW_t\right)^2\right] = \mathbb{E}\left[\left(\int_0^T X_t^2 dt\right)\right] \tag{3.53}$$

where W_t is a Wiener process and X_t is a stochastic process that is adapted to the filtration $\mathcal{F}$ of the Wiener process.

have

$$
\begin{aligned}
cov &\left[\int_0^\tau f(i,k)e^{-s_k(t_1-v)}dw_v^{(l)}, \int_0^\tau f(j,m)e^{-s_m(t_2-v)}dw_v^{(l)}\right] \\
&= \int_0^\tau \sum_{l,k,m} \left((\mathbf{U}^{-1})_{i,k}(\mathbf{U}\mathbf{\Lambda})_{k,l}e^{-s_k(t_1-v)}\right)\left((\mathbf{U}^{-1})_{j,m}(\mathbf{U}\mathbf{\Lambda})_{m,l}e^{-s_m(t_2-v)}\right)dv \\
&= \sum_{l,k,m}(\mathbf{U}^{-1})_{i,k}(\mathbf{U}\mathbf{\Lambda})_{k,l}(\mathbf{U}^{-1})_{j,m}(\mathbf{U}\mathbf{\Lambda})_{m,l}\left(e^{-s_kt_1-s_mt_2}\int_0^\tau e^{(s_k+s_m)v}dv\right).
\end{aligned}
$$

Finally, without loss of generality let $t_2 > t_1$ and write $t_2 = t_1 + \Delta t$. Then consider the last integral term above in brackets,

$$
\begin{aligned}
&\left(e^{-s_kt_1-s_mt_2}\int_0^\tau e^{(s_k+s_m)v}dv\right) \\
&\quad = e^{-(s_k+s_m)t_1-s_m\Delta t}\int_0^{t_1} e^{(s_k+s_m)v}dv \\
&\quad = e^{-(s_k+s_m)t_1-s_m\Delta t}\left(\frac{e^{(s_k+s_m)t_1}-1}{s_k+s_m}\right) \quad \xrightarrow{t_1\to\infty} \quad \frac{e^{-s_m\Delta t}}{s_k+s_m}.
\end{aligned} \tag{3.54}
$$

Thus, the kernel function is

$$
\begin{aligned}
k\left[x_{t_1}^{(j)}, x_{t_2}^{(i)}\right] &= cov\left[x_{t_1}^{(j)}, x_{t_2}^{(i)}\right] \\
&= \begin{cases} \sum_{l,k,m}(\mathbf{U}^{-1})_{i,k}(\mathbf{U}\mathbf{\Lambda})_{k,l}(\mathbf{U}^{-1})_{j,m}(\mathbf{U}\mathbf{\Lambda})_{m,l}\frac{e^{-s_m\Delta t}}{s_k+s_m} & , t_2 > t_1 \\ \sum_{l,k,m}(\mathbf{U}^{-1})_{i,k}(\mathbf{U}\mathbf{\Lambda})_{k,l}(\mathbf{U}^{-1})_{j,m}(\mathbf{U}\mathbf{\Lambda})_{m,l}\frac{e^{-s_k\Delta t}}{s_k+s_m} & , t_1 > t_2 \end{cases}.
\end{aligned} \tag{3.55}
$$

3.5 Learning and Inference

3.5.1 Learning

In this section, we first focus on the learning of univariate KCPs. Then using the univariate KCPs learning as the building block, we will extend the approach to the multivariate KCPs.

The learning of univariate KCPs can be performed by maximizing likelihood. The likelihood function of a generic KCP is given by:

$$
\mathcal{L}(\theta) = \mathbb{P}(\mathbf{y}|\mathbf{x},\theta) = \mathfrak{c}(\{F_i(y_i)\}_{i=1}^N)\prod_{i=1}^N f_i(y_i), \tag{3.56}
$$

where $\mathbf{y}$ is the set of training data and θ is the set of hyper-parameters of the copula process.

For instance, assuming a Gaussian copula and Student's t marginals, the negative log-likelihood function is

$$-\log(\mathcal{L}(\theta)) = \frac{1}{2}\log|\Lambda| + \frac{1}{2}\mathbf{z}^T\Lambda^{-1}\mathbf{z} - \frac{1}{2}\mathbf{z}^T\mathbf{z} - \sum_{i=1}^{N}\log(t(y_i,\nu)) \tag{3.57}$$

where $\mathbf{z}^T = [z_1 = \Phi^{-1}(T_\nu(y_1)); \ldots; z_N = \Phi^{-1}(T_\nu(y_N))]$ and $T_\nu(y_i)$ and $t_i(y_i,\nu)$ are the univariate Student's t-distribution and density functions (with degree of freedom ν) respectively; and θ are the hyper-parameters of the kernel function. Here, without loss of generality the time-series samples are assumed to be standardized. Bayesian learning can be performed by specifying prior distributions for the parameters and integrate them out (for examples in the GP context, see [85]). Computational time will suffer as a result. However, this extra computation can be avoided safely if the initial values for the parameters are chosen well based on observations from the data. We will discuss such an initialization method in Chapter 4.

Training becomes more interesting for the multivariate KCP models in Equation (3.32). There are two ways to train these models. First, a model can be trained in a single step where all the parameters of the univariate KCPs and the binding copula are learned at the same time. The likelihood function in this case is identical to Equation (3.32). However, this straightforward approach tends to be computationally slow as the gradient of all parameters must be computed numerically in every step, involving the entire multi-series data set. This problem is compounded by the fact that multiple random-restarts[8] will likely be necessary to obtain likelihoods close to global maximum. Another computationally faster approach is to follow a two-step process when the univariate KCPs for each series are first learned independently. Then the parameters of the binding copula are learned while holding the univariate KCPs parameters at the

[8]The term *random-restart* refers to the operation where MLE is repeated from scratch multiple times using randomly generated initial parameter values.

respective maximum likelihood values. The two-stage modular learning proposed here is similar in spirit to the Inference Function for Margins (IFMs) by Joe and Xu [52] and Maximization by Parts (MBPs) by Song *et al.* [103]. Specifically in the case of the multivariate KCPs, the operation of learning the binding copula in the second stage is no different from the learning for a copula in an ordinary case involving simple random variables. To see this, we again consider the likelihood function Equation (3.32) below, while explicitly showing the parameter dependencies:

$$\begin{aligned}&\mathcal{L}_{MV}\left(\theta_b, \{\theta_k\}_{k=1}^{D}\right)\\ &\quad= \prod_{t'=1}^{t}\left\{\mathfrak{c}_b\left(\left\{p_{t'}^{(k)}\right\}_{k=1}^{D};\theta_b\right)\cdot\left[\prod_{k=1}^{D}\left(\frac{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'};\theta_k\right)}{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'-1};\theta_k\right)}\cdot f_k(y_{t'}^{(k)};\theta_k)\right)\right]\right\}\end{aligned} \tag{3.58}$$

Notice the terms inside the square bracket in Equation (3.58) have constant values given that the parameters of the univariate KCPs $\{\theta_k\}_{k=1}^{D}$ are held at the likelihood maximizing values. Thus, these terms can be removed from consideration during the MLE of the binding copula parameters θ_b, and the likelihood function in Equation (3.58) simplifies to a likelihood function of a static copula:

$$\mathcal{L}_{MV}\left(\theta_b\right)|_{\{\theta_k\}_{k=1}^{D}} \propto \prod_{t'=1}^{t}\mathfrak{c}_b\left(\left\{p_{t'}^{(k)}\right\}_{k=1}^{D};\ \theta_b\right). \tag{3.59}$$

Such simplification significantly reduces the computational speed. Moreover, in the two-stage estimation method, the univariate KCPs are trained separately, which lends itself naturally to parallel computing.

Further computational savings can be had in the computations of the conditional probabilities $p_t^{(k)}$ by limiting the size of the set of observations that are being conditioned

on using a sliding window:

$$\begin{aligned} p_t^{(k)} &= \mathbb{P}\left(Y_t^{(k)} < y_t^{(k)} \;\middle|\; \left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=1}^{t-1}\right) \\ &= \frac{\mathbb{P}\left(\left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=1}^{t}\right)}{\mathbb{P}\left(\left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=1}^{t-1}\right)} \\ &\sim \frac{\mathbb{P}\left(\left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=t-N_w-1}^{t}\right) \cdot \mathbb{P}\left(\left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=1}^{t-N_w}\right)}{\mathbb{P}\left(\left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=t-N_w-1}^{t-1}\right) \cdot \mathbb{P}\left(\left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=1}^{t-N_w}\right)} \\ &= \mathbb{P}\left(Y_t^{(k)} < y_t^{(k)} \;\middle|\; \left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=t-N_w-1}^{t-1}\right). \end{aligned} \tag{3.60}$$

The approximation in the third line of Equation (3.60) holds when kernel functions with some type of bandwidth or decaying parameter are used, and when the sliding window size is chosen to be much greater than the bandwidth parameter. For example, when an OU kernel (Equation (3.5)) is used, the Gram matrix is mostly diagonal as shown in Figure 3.8. That is, the Gram matrix is only substantially non-zero near the main diagonal. This means that the data points over a certain distance away from the point of interest have virtually no effect on the corresponding predictive distribution. Thus the joint distribution $\mathbb{P}\left(\left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=1}^{t}\right)$ approximately factors into $\mathbb{P}\left(\left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=t-N_w-1}^{t}\right) \cdot \mathbb{P}\left(\left\{Y_i^{(k)} = y_i^{(k)}\right\}_{i=1}^{t-N_w}\right)$. As such, the use of a sliding window for model training is justified, as long as the training window size is much larger than the bandwidth parameter. This feature of the Gram matrix is in fact quite common for a lot of kernel functions (e.g. the RBF kernel and the heteroskedastic kernel in Equations (3.4) and (3.41)) as the typical assumption is that data points closer in feature space tend to be more similar. This proposed sliding window scheme will be used in Chapter 4 when applying the KCP models to real-life applications.

The modular nature of the learning procedure above also lends itself well to applications such as portfolio management, where different assets are regularly added and removed. In such applications, the individual time-series, representing asset prices or

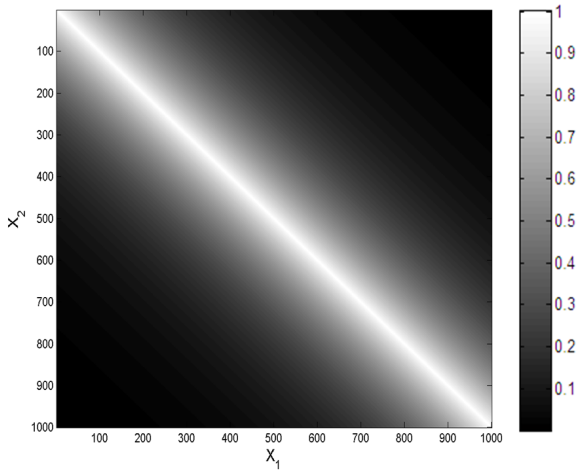

Figure 3.8: The Gram matrix of the OU kernel. The fact that non-zero entries are concentrated near the main diagonal justifies the use of a sliding window scheme for model training in order to realize more computational savings.

associated economic indicators, can be modeled independently from each other. Thus addition or removal from the overall models only triggers the relearning of the binding copula without propagating to the rest of the models. In a portfolio that potentially contains several hundreds or thousands of names[9], this property can translate into significant savings in computational time and energy bills.

3.5.2 Inference

One of the strengths of KCPs (which is also enjoyed by GPs) is that the entire predictive distribution is available during inference, thus predictions are not limited to point prediction, which allows for much more accurate risk management.

[9]For example, the NYSE has a listing of 2,773 stocks and NASDAQ has a listing of 3,800+ stocks as of end of 2009.

The predictive distribution and density of a univariate KCP are given by:

$$\mathbb{P}(\mathbf{Y}^* \leq \mathbf{y}^* | \mathbf{Y} = \mathbf{y}) = \frac{\mathbb{C}_{\tilde{\Lambda}}(\{F_i(y_i)\}_{i=1}^N, \{F_j(y_j^*)\}_{j=1}^M)}{\mathbb{C}_{\Lambda}(\{F_i(y_i)\}_{i=1}^N)}, \quad \text{and} \tag{3.61}$$

$$\mathbb{P}(\mathbf{y}^* | \mathbf{y}) = \frac{\mathfrak{c}_{\tilde{\Lambda}}(\{F_i(y_i)\}_{i=1}^N, \{F_j(y_j^*)\}_{j=1}^M)}{\mathfrak{c}_{\Lambda}(\{F_i(y_i)\}_{i=1}^N)} \cdot \prod_{j=1}^{M} f_j(y_j^*), \tag{3.62}$$

where Λ and $\tilde{\Lambda}$ are the Gram matrices of the observations and the combined observations and targets respectively as defined in Equation (3.3). The dependency on $\mathbf{x}$ and $\mathbf{x}^*$ is implicit and it is dropped from the notation for clarity of presentation.

The *maximum a-posterior* (MAP) estimator could be used to provide a point estimate of the target values:

$$\bar{\mathbf{y}^*} = \max_{\mathbf{y}^*} \mathbb{P}(\mathbf{y}^* | \mathbf{y}). \tag{3.63}$$

When the choice of copula functions and marginal distributions result in a symmetric predictive distribution, the MAP estimate will coincide with the mean-squared estimator $\mathbb{E}[\mathbf{y}^* | \mathbf{y}]$.

3.5.3 Sample Path Generation

The generation of sample paths in the univariate KCP setting is very simple and similar to the procedure outlined for ordinary copula functions provided by Cherubini *et. al.* [17]. For example, generating a sample path from a univariate KCP with the Student's t-copula with a kernel function $k(x_i, x_j)$, degree of freedom ν_c, and marginal distribution $F(y)$, proceeds as follows:

1. compute the Gram matrix Λ, with $\Lambda_{i,j} = k(x_i, x_j)$ for $i, j = 1, \ldots, N$;
2. compute the Cholesky decomposition $\mathbf{L}$, such that $\Lambda = \mathbf{L}\mathbf{L}^T$;
3. sample N independent draws $\widehat{\mathbf{v}} = [\widehat{v}_1, \ldots, \widehat{v}_N]^T$ from $\mathcal{N}(0, 1)$;
4. sample a random number s from a Chi-square distribution $\mathcal{X}^2_{\nu_c}$, independent of $\widehat{\mathbf{v}}$;

5. compute $\widehat{\mathbf{w}} = \mathbf{L}\widehat{\mathbf{v}}$ (generates correlated random samples);

6. compute $\widehat{\mathbf{z}} = \sqrt{\frac{\nu_c}{s}}\widehat{\mathbf{w}}$ (generates correlated Student's t distributed samples);

7. compute $\widehat{u}_i = T_{\nu_c}(\widehat{z}_i)$ for $i = 1, \ldots, N$, where again $T_{\nu_c}(\cdot)$ denotes the standard univariate Student's t CDF (generates correlated uniformly distributed samples); and

8. Finally, obtain the sample path by $\widehat{\mathbf{y}} = F^{-1}(\widehat{\mathbf{u}})$.

With the above procedure, all the samples of the entire univariate sample path can be generated at the same time. The multivariate KCP in Equation (3.32), on the other hand, must be generated using an iterative method. A procedure similar to the univariate KCP sampling can be used. However, instead of sampling from an elliptical copula in the serial (or time) dimension, we use the procedure to generate samples from a binding copula in the cross-sectional sense.

Using the same notations as in Section 3.2, a set of $\left\{\widehat{p}_t^{(k)}\right\}$ for $k = 1, \ldots, D$ can be generated by executing steps 1 to 7 of the above procedure using a binding copula, instead of the elliptical copula as in the univariate case. The process is repeated for each time step $t = 1, \ldots, N$. The result is a set of cross-sectional samples (indexed by k) of $\left\{\widehat{p}_t^{(k)}\right\}$, which are connected at time t by the binding copula. Note that the sets of samples across time remain independent, i.e. $\left\{\widehat{p}_t^{(k)}\right\}$ and $\left\{\widehat{p}_s^{(k)}\right\}$ are independent for $t \neq s$. To this end, the temporal dependency is constructed by passing these sets of samples through the inverse of Equations (3.20) to (3.22), starting from $t = 1$ using

Equation (3.21) as follows:

$$\widehat{y}_1^{(k)} = F_k^{-1}\left(\widehat{p}_1^{(k)}\right), \tag{3.64}$$

$$\widehat{y}_2^{(k)} = F_k^{-1}\left(g_k^{-1}\left(\widehat{p}_2^{(k)}\right)\Big|_{\widehat{u}_1^{(k)}}\right), \tag{3.65}$$

$$\widehat{y}_3^{(k)} = F_k^{-1}\left(g_k^{-1}\left[\widehat{p}_3^{(k)} \cdot \mathfrak{c}_k\left(\widehat{u}_2^{(k)}, \widehat{u}_1^{(k)}\right)\right]\Big|_{\widehat{u}_2^{(k)}, \widehat{u}_1^{(k)}}\right), \tag{3.66}$$

$$\cdots$$

$$\widehat{y}_N^{(k)} = F_k^{-1}\left(g_k^{-1}\left[\widehat{p}_N^{(k)} \cdot \mathfrak{c}_k\left(\left\{\widehat{u}_t^{(k)}\right\}_{t=1}^{N-1}\right)\right]\Big|_{\left\{\widehat{u}_t^{(k)}\right\}_{t=1}^{N-1}}\right). \tag{3.67}$$

In this manner, sample paths from multivariate KCPs with arbitrary kernel functions, copulae, and marginal distributions can be generated. On the computation of $g_k^{-1}(\cdot)$, given $g_k^{-1}(\cdot)$ is monotonic and typically smooth given the use of elliptical copulae and continuous marginals in univariate KCPs, the computation requires a simple application of importance sampling [76] [62]. In the case of a KCP with a Gaussian copula, the computation of $g_k^{-1}(\cdot)$, can further simplify by inverting Equation (3.23) directly as follows:

$$g_k^{-1}\left(p_t^{(k)}\right)\Big|_{\left\{u_i^{(k)}\right\}_{i=1}^{t-1}} = \Phi_{\mu;\Sigma}^{-1}\left(\frac{p_t^{(k)} \cdot \mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t-1}\right)}{\mathfrak{c}_{\mathbf{A}}(u_1, \ldots, u_{t-1})}\right), \tag{3.68}$$

where the same notations have been carried over from Equation (3.23).

3.6 A Note on Missing Data

Missing data is an inevitable fact of life that happens for many reasons. Data points can be lost in transmission, storage corruption, or simply lost because of equipment or human errors, just to name a few. In the context of this work, *missing data* refers to the missing samples on an otherwise regularly sampled time-series. Missing data disrupts the direct application of many existing models. For example in the ARMA model, the variance at each time step is computed from the variance and observations from the previous time step.

There are two main ways to handle missing data: 1) filling in data values; 2) incorporating the missing data as a latent variable in the model directly. Typically data-filling is performed by some interpolation methods. The complexity and effectiveness varies with the method chosen. It can be as simple as using linear interpolation on the data points near the missing data point, or as complex as some nonlinear techniques such as Singular Spectrum Analysis (SSA) [46], which creates a probable sample of the missing data by taking into account trends in different scales and periodicities with varying amplitudes and different frequencies. Alternately, for probabilistic models, one can treat missing data as latent variables in the model. This is the case for the HMM, where the estimation of the missing data is incorporated directly in the forward-backward recursion.

In any case, the handling of missing data can be daunting and tedious for most models, adding unwanted pre-processing or complexity to the model. To this end, the KCPs provide an effortless way to accommodate missing data: with no pre-processing nor added complexity.

First consider the case of univariate KCP with no missing data. To make predictions at $t+1$ with a KCP with learned model parameters θ and observations $\{y_t, \ldots, y_0\}$, one simply computes the predictive distribution $\mathbb{P}(y_t|\{y_{t-1}, \ldots, y_0\})$ (See Equation (3.62)). When one or more data points are missing, the historic data set that the predictive distribution is conditioned on is simply reduced and the predictive distribution to be computed becomes $\mathbb{P}(y_t|\mathbf{y}_\pi)$, where the $\mathbf{y}_\pi$ is the set of available historic data. Specifically, if the observation y_l is missing, then the predictive distribution becomes

$$\begin{aligned}\mathbb{P}(y_t|y_{t-1}, \ldots, y_{l+1}, y_{l-1}, \ldots, y_0) &= \frac{\mathfrak{c}(u_t, \ldots, u_{l+1}, 1, u_{l-1}, \ldots, u_0)}{\mathfrak{c}(u_{t-1}, \ldots, u_{l+1}, 1, u_{l-1}, \ldots, u_0)} \cdot f(y_t) \\ &= \frac{\mathfrak{c}(u_t, \ldots, u_{l+1}, u_{l-1}, \ldots, u_0)}{\mathfrak{c}(u_{t-1}, \ldots, u_{l+1}, u_{l-1}, \ldots, u_0)} \cdot f(y_t), \quad (3.69)\end{aligned}$$

where setting u_l to 1 is equivalent of integrating the missing variable y_l out. The remainder of the inference operation is unchanged. This is also true for learning as well, making the handling of missing data effortless.

The handling of missing data in the multivariate KCP is also effortless. If the observation $y_l^{(d)}$ is missing, one can simply set the corresponding missing data $p_l^{(d)}$ in Equation (3.32) to 1 and omit the corresponding term in the second product of Equation (3.32):

$$\begin{aligned}\mathbb{P}\left(\{\mathbf{Y}_i = \mathbf{y}_i\}_{i=1}^{t}\right) = & \prod_{t' \in \{1,\ldots,l-1,l+1,\ldots,t\}} \left[\mathfrak{c}_b\left(p_{t'}^{(1)}, \ldots, p_{t'}^{(D)}\right) \cdot \prod_{k=1}^{D} \left(\frac{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'}\right)}{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'-1}\right)} \cdot f_k(y_{t'}^{(k)}) \right) \right] \cdot \\ & \mathfrak{c}_b\left(p_{t'}^{(1)}, \ldots, p_{t'}^{(d-1)}, 1, p_{t'}^{(d)}, \ldots, p_{t'}^{(D)}\right) \cdot \\ & \prod_{k \in \{1,\ldots,d-1,d+1,\ldots,D\}} \left(\frac{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'}\right)}{\mathfrak{c}_k\left(\left\{u_i^{(k)}\right\}_{i=1}^{t'-1}\right)} \cdot f_k(y_{t'}^{(k)}) \right)\end{aligned} \tag{3.70}$$

Setting the missing $p_l^{(d)}$ terms to 1 is equivalent to integrating out the missing data. This is a direct consequence of using the binding copula function in the multivariate KCP formulation, that is

$$\mathbb{C}_b\left(p_1, \ldots, p_{i-1}, 1, p_{i+1}, p_N\right) = \mathbb{C}'_b\left(p_1, \ldots, p_{i-1}, p_{i+1}, p_N\right), \tag{3.71}$$

where the copula function $\mathbb{C}'_b(\cdot)$ has a dimension one lower than $\mathbb{C}_b(\cdot)$. In the multivariate KCP framework, the PIT sequences are uniformly distributed. Thus marginalizing out a particular random variable is equivalent to setting $p_l^{(d)} = 1$.

3.6.1 Synthetic Example - Missing Data

In this section, the synthetic example introduced in Section 3.1.3 is extended to illustrate the procedure using KCPs for handling missing data. The same AR(2) process in Equation (3.16) is used as the underlying DGP. Similar to the example given in Section 3.1.3, the predictive distribution at $t^* = 5$ is computed using a GCTM-KCP. The data points $\{y_t\}, t \in [-5, 5)$ depicted in Figure 3.5 are used as training data $\mathcal{D}$, i.e. $\mathcal{D} = \{t, y_t\}, t \in [-5, 5)$. The performance of the GCTM-KCP (with an OU kernel) can be further assessed under missing data conditions by removing data points from the

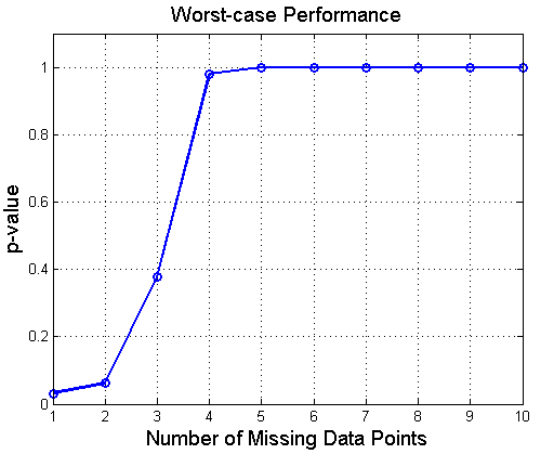

Figure 3.9: p-values of the $\mathcal{X}^2$ tests performed under progressively severe missing data conditions.

training set. Data points starting from the one closest to $t^* = 5$ are removed one at a time. Each time a data point is removed from the training set, a new set of model parameters is learned, and a new predictive distribution at $t^* = 5$ is computed. This process is repeated a total of ten times. As explained later in this section, this test constitutes a worst-case performance test for the GCTM-KCP under missing data conditions.

The training of the GCTM-KCP model is performed with MLE as described in Section 3.5.1. The predictive distribution are computed as described in Equation (3.6).

Recall that the true posterior distribution for the AR(2) process is known at $t = 5$ since the DGP is known. Thus the difference between the true posterior distribution and the generated set of predictive distributions can be evaluated using the $\mathcal{X}^2$ test. The p-values of this series of tests are plotted in Figure 3.9.

Recall that in the $\mathcal{X}^2$ test, a p-value smaller than $\alpha \in [0, 1]$ means the null hypothesis (stating that the two test distributions are the same) is accepted with $(1 - \alpha)$ significance level. In the context of this experiment, the smaller the p-value, the better the GTCM-KCP is performing. As observed in Figure 3.9, the performance of the GTCM-KCP degraded dramatically when there are more than two missing data points. This observation can be explained by the fact that when a test point t^* is sufficiently far

away (as compared to the length-scale parameter h) from all the other training points, the predictive distribution $\mathbb{P}(y^*|t^*, \mathcal{D})$ approaches to the unconditional marginal distribution. In fact, the rate of relaxation to the unconditional marginal is exponential due to the exponentiation in the OU kernel function. To understand this, first consider the joint distribution of a pair of training and test points $\mathbb{P}(y^*, \mathbf{y}|t^*, \mathbf{t})$:

$$\mathbb{P}(y^*, \mathbf{y}) \to \mathbb{P}(y^*) \cdot \mathbb{P}(\mathbf{y}) \text{ as } |t^* - t_0| \to +\infty \text{ since } k_{OU}(t^*, t_0) \to 0, \tag{3.72}$$

where t_0 is the location of the training data point closest to the test point t^*. Thus the posterior distribution $\mathbb{P}(y^*|t^*, \mathcal{D}) \to \frac{\mathbb{P}(y^*)\cdot\mathbb{P}(\mathbf{y})}{\mathbb{P}(\mathbf{y})} = f(y)$, yielding the unconditional marginal distribution.

The closer a training data point y_{t_0} is to a test point y^* at t^*, the more influential it is on the predictive distribution $\mathbb{P}(y^*|t^*, \mathcal{D})$ (because the value $k_{OU}(t^*, t_0)$ is higher as $|t^* - t_0|$ gets smaller). Therefore, by first removing the data points closer to the test point in this experiment, the more significant data points are removed first. This facilitates a faster degradation of the KCP performance than if other data points were removed first. Thus, the above experiment represents the worst-case performance of the KCPs to missing data.

3.7 Model Selection and Performance Metrics

In regression and out-of-sample prediction of stochastic processes, the goal is to obtain an accurate predictive distribution. Point-predictions are meaningless since the realized value from a stochastic process will almost always differ from the prediction. Knowing the entire predictive distribution on the other hand yields a much more complete picture. As such, conventional performance metrics such as least-square errors between the point-predictions and the set of realized values is of little use, which departs from the typical metrics used in the time-series analysis community [65].

To this end, the model selection and ranking metrics used in this work will be focused on two categories: *in-sample* and *out-of-sample* or *predictive*. Please note, the categorization used here may be somewhat different from their conventional use. The likelihood function and its derivatives such as the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are considered to be *in-sample* as their values are solely computed based on the training data. The probability integral transforms (PIT) [22] and predictive likelihoods are considered to be *out-of-sample* or *predictive* as they are designed to directly evaluate the power of the predictive densities generated by each competing model.

First, consider the *in-sample* performance metrics. The likelihood functions of the KCPs and other models can be found throughout this chapter and will not be repeated here. The definitions of AIC and BIC are given below:

$$AIC = -2\ln\mathcal{L} + 2k, \tag{3.73}$$

$$BIC = -2\ln\mathcal{L} + k\ln N, \tag{3.74}$$

where $\mathcal{L}$ is the maximized log-likelihood, k is the number parameters in a model, and N is the number of data points used in computing the likelihood. Numerically, the only difference between AIC and BIC is that BIC penalizes the complexity of the model more heavily. Further, unlike the likelihood ratio, the models under consideration do not have to be nested, thus making them ideal for ranking a wide variety of models. Given the likelihood functions, both the AIC and BIC are simple to compute. There are a few theoretical justifications for using the AIC [58]. Firstly, Shibata [96] pointed out that the AIC picks the correct model asymptotically, in the Kullback-Leibler sense, if the complexity of the true model grows with sample size. For instance, this is the case if the true data generating process is autoregressive. Secondly, when precision of the prior is comparable to that of the likelihood, model comparisons based on the AIC are asymptotically equivalent to those based on Bayes factors [2]. In the more typical

situation where information in the prior is small relative to the information provided by the data, Schwarz [95] showed that the model with the highest posterior probability is the one that minimizes the BIC in Equation (3.74).

There are criticisms of both measures. The AIC and BIC do not incorporate uncertainty about the parameter values and model form [1]. Shibata [96] and Katz [59] have also shown that the AIC tends to overestimate the number of parameters needed, even asymptotically. The BIC further assumes that data points are *iid*, which is not typically the case in time-series analysis. Thus, complementary criteria to investigate the predictive powers of each model is needed.

In the case of measuring the goodness-of-fit modeling stationary random variables, the *quantile-quantile* plot (or commonly known as q-q plot) serves as a good graphical tool for comparing the distributions of the samples drawn from a model and the true data distribution [39]. The q-q plot is especially useful in showing any mismatch at the tail ends of the model distribution. However, the validity of the use of the q-q plot does not carry over to stochastic processes and time-series models as most time-series can only be observed once. In the case of time-series forecasting, for instance, a predictive distribution is produced by the model at the time of interest. In order to construct a q-q plot, multiple samples from the true data generating process must be available, and this is not possible in most practical applications. Furthermore, even in the rare cases of repeatable experiments where a stochastic process or time-series can be observed repeatedly, the goodness-of-fit inferred from the q-q plot cannot be quantified, thus making it unsuitable for large scale experiments.

To this end, the PIT is employed as the complementary measure of the predictive power of models. The PIT has been used in the context of goodness-of-fit tests as far back as the 1930s; see Pearson [84] for example. More recently, Diebold *et. al.* [22] extended the PIT theory to the time-series case and proposed using it in the evaluation of density forecasts. While the maximized likelihood relates to the predicted density having

minimum variance about the observations[12], PIT measures how well the predictive distribution generated by a particular model corresponds to the distribution of the true data generating process. The PIT of a particular model is simply the cumulate density function corresponding to the predictive density of the model evaluated at realization of the actual process y_t:

$$p_t = \int_{-\infty}^{y_t} \mathbb{P}(\nu|\widetilde{\mathcal{D}}, \theta, \mathcal{M}_i)d\nu, \tag{3.75}$$

where $\widetilde{\mathcal{D}} = \{y_j\}_1^{t-1}$ is the training data, while θ is the parameter set learned under the model assumptions of $\mathcal{M}_i$. Note that to speed up computations, the predictive distribution can be computed using a reduced training set with a sliding window of width N_w, i.e. $\widetilde{\mathcal{D}} = \{y_j\}_{t-N_w-1}^{t-1}$, as long as N_w is chosen to be much greater than the bandwidth parameter of the kernel function.

As was shown in Diebold [22], the PIT series $\{p_t\}$ should be *iid* uniform if the model $\mathcal{M}_i$ predicts the true underlying data generating process (DGP) well. Thus inspecting the histogram and the ACF of the PIT series $\{p_t\}$ serves as an evaluation of the goodness-of-fit of a given model. Conversely, by visually examining the shape of the PIT series histogram, one can diagnose the misspecification of the predictive distribution generated by a model as shown in Figure 3.10.

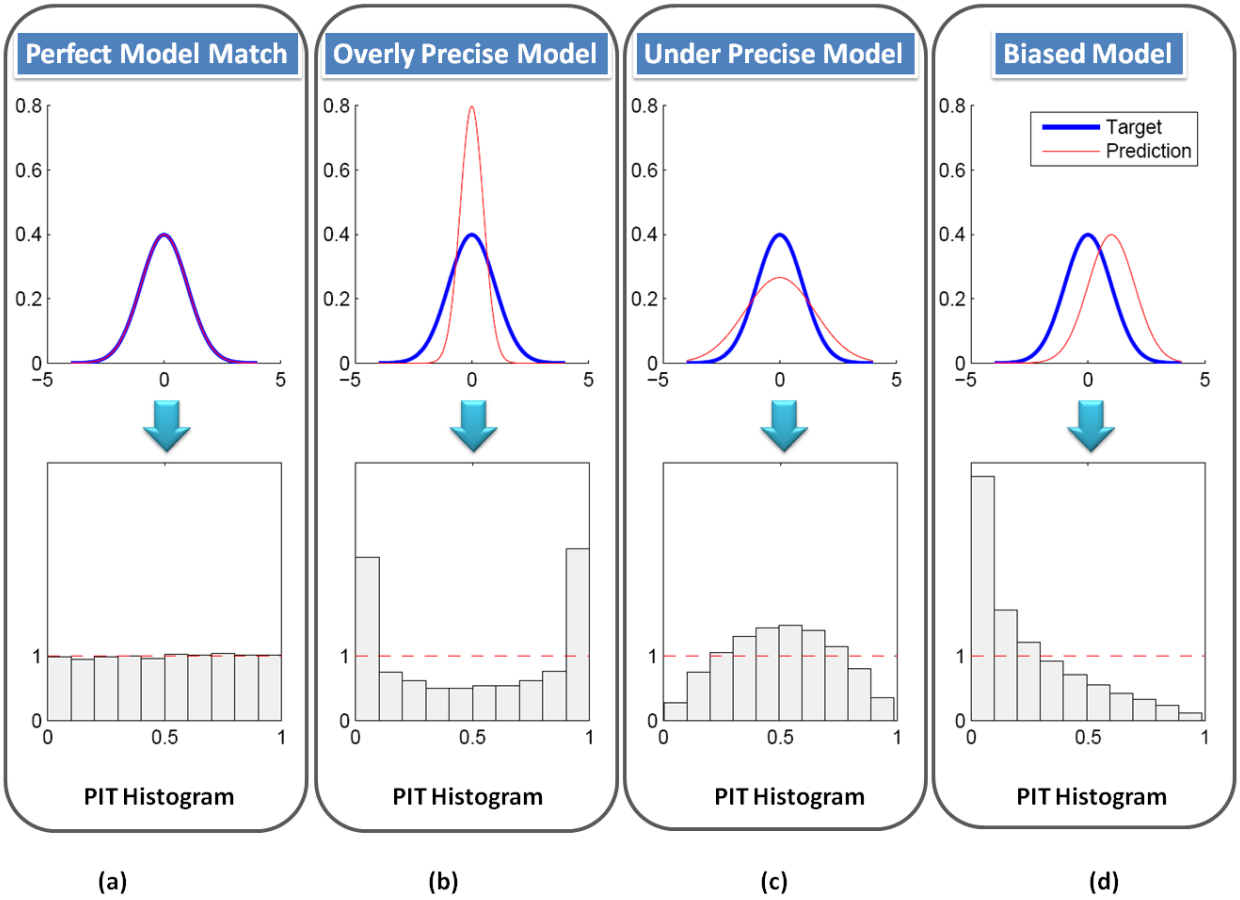

Figure 3.10: Possible scenarios when evaluating goodness-of-fit with PIT. (a): The predictive distribution learned by the model matches the distribution of the true underlying data generation process (DGP). A perfect uniform distribution results when considering the histogram of the PIT series; (b): The model is unbiased, but it is too precise compared to the true DGP; (c): The model is unbiased, but it is under-precise compared to the true DGP; (d) The model precision is just right, but it is biased. Of course, a model can be biased and over- or under-precise at the same time.

Table 3.2: Critical Values for the Anderson-Darling Criterion.

Significance Levels	Critical Values
0.10	1.933
0.05	2.492
0.01	3.857

In addition to this graphical evaluation approach, a further benefit of using the PIT is that the goodness-of-fit can be easily quantified with the Anderson-Darling criterion (ADC):

$$W_n^2 = -n - \frac{1}{n}\sum_{j=1}^{n}(2j-1)[\ln u_j + \ln(1-u_{n-j+1})], \tag{3.76}$$

where the $\{u_j\}$ are the sorted version (ascending order) of the PIT series $\{p_t\}$ and n is the number of entries in $\{p_t\}$. The ADC has the added advantage that it produces a score that focuses goodness-of-fit at the tail regions of the distributions [4], which is often the focus of financial and actuarial applications. According to the ADC, the hypothesis of $\mathcal{M}_i$ being the correct model must be rejected if this score is too high. The critical values corresponding to different significance levels are listed in Table 3.2 [4].

3.8 Comparison with Other Competing Models

There are four main contenders for the KCPs. These are the GARCH model[25][9], the Gaussian processes[85], the warped Gaussian processes[101], and dynamic copula[83]. A list of merits of each model was provided in Chapter 2. This section will contrast the distinct contributions provided by the KCP model.

GARCH is a finite memory model: its memory and modeling power resides in the first two moments. KCP's long-term memory can be embedded in a more complex way through the use of the copula function. Further, the model complexity of GARCH increases linearly with the range of memory of the model. One major limitation is that the GARCH model assumes data comes in regular intervals, an assumption that is not true in most real-life situations. Even in *daily* financial data, often data is not available on weekends and holidays due to market inactivity. The rise of high-frequency finance further exposes this weakness as transactions happen at random times. The KCP has no such restrictions.

The Gaussian process is a special case of KCP. The main benefit of GP is its high relative computation efficiency. However, it suffers from the rigid and unrealistic assumption of normality. Snelson's warped GP [101] is the latest extension of GP to relax the normality assumption via nonlinear transformation. Many constraints are imposed on the selection of warping functions. For example, a warping function must be monotonic, i.e. invertible, with a readily computable gradient for learning. There is neither a clear guideline nor logical method for choosing the warping function which matches the observed distributions. Above all, the warping function is fixed once it is chosen, thus all subsequent predictions are a constant transformation from the Gaussian distribution. The KCP is much more flexible in this regard as the predictive distribution is completely driven by the training data.

Patton's dynamic copula [83] attempts to capture the dynamics among time-series, while the KCPs capture the dynamics in the univariate time-series level before connecting

time-series through a binding copula. Patton's approach simply represents a different model choice. However, the dynamic copula model provides no clear guidelines for or justification of how to design the evolution equations for the copula parameters. As a result, the application of the model has so far been limited to the modeling of currency rates as was originally proposed by the authors. The KCPs on the other hand provide a set of clear design guidelines and the same evolution equations can be readily adopted. Most importantly, the KCPs are applicable to various domains as will be illustrated throughout in this work.

3.9 KCP Classification

KCP classification follows the *discriminative* approach, where the posterior distribution $\mathbb{P}(y = \mathcal{C}_k|\mathbf{X})$ of class labels $\{\mathcal{C}_k\}_{k=1}^K$ given the feature space $\mathbf{X} \in \mathcal{R}^D$, is modeled directly using a KCP. The process of learning such a model is almost identical to the process of KCP regression presented, with one important twist. For simplicity, first consider the binary classification problem, where without loss of generality the class labels are denoted as $\{0, 1\}$. The problem of predicting the class label can be treated as a regression problem if the observed class labels are considered as functions of the features, i.e. $y = h(\mathbf{x})$. The major difference in this case is that the observed values of y will be concentrated at 0 and 1. As a result, consider the following bi-modal marginal distribution, which is a mixture of two Gaussians:

$$F_{MoG}(y) = p \cdot \Phi(y; 0, \xi) + (1 - p) \cdot \Phi(y; 1, \xi), \tag{3.77}$$

$$f_{MoG}(y) = p \cdot \phi(y; 0, \xi) + (1 - p) \cdot \phi(y; 1, \xi), \tag{3.78}$$

where p is a constant between 0 and 1, $\Phi(y; \mu, \xi)$ and $\phi(y; \mu, \xi)$ are, as usual, the Gaussian CDF and the PDF with mean μ and variance ξ. The variance ξ can be fixed and must be chosen such that given the numerical representations of the class labels, the individual Gaussian distributions have no overlap. Its exact value is not very important as there

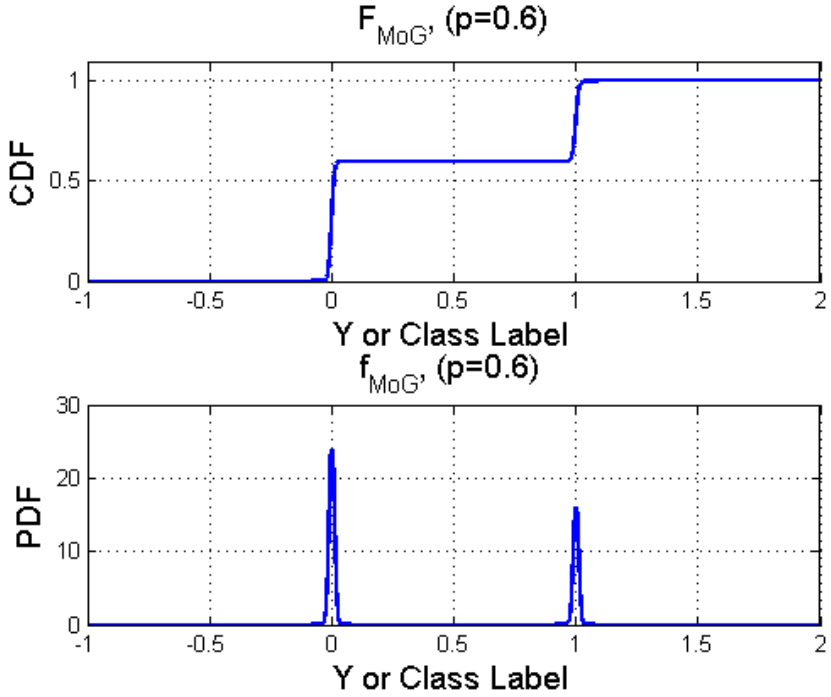

Figure 3.11: The bimodal distribution – a candidate marginal distribution for binary classification. A mixture of two Gaussians is used here as an example with $p = 0.6$ and $\xi = 0.01$ (Equation 3.78).

will be no observed values besides the class labels. Figure 3.11 depicts an example with $p = 0.6$ and $\xi = 0.01$.

The remaining procedure for KCP classification is now exactly the same as KCP regression. Specifically, the joint distribution of the *training* (observed) class labels $\mathbf{y} = \{y_1, \ldots, y_N\}$ and *test* class labels (prediction targets) $\mathbf{y}^* = \{y_1^*, \ldots, y_M^*\}$ given the respective features $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and $\mathbf{X}^* = \{\mathbf{x}_1^*, \ldots, \mathbf{x}_M^*\}$ can be written as

$$\mathbb{P}(\mathbf{Y} \leq \mathbf{y}, \mathbf{Y}^* \leq \mathbf{y}^* | \mathbf{X}, \mathbf{X}^*) = \mathbb{C}(\{F_{MoG}(y_i)\}_{i=1}^N, \{F_{MoG}(y_j^*)\}_{j=1}^M | \mathbf{X}, \mathbf{X}^*) \tag{3.79}$$

and the joint density can once again be obtained by differentiating with respect to $\{\mathbf{y}, \mathbf{y}^*\}$:

$$\mathbb{P}(\mathbf{y}, \mathbf{y}^* | \mathbf{X}, \mathbf{X}^*) = \mathfrak{c}(\{F_{MG}(y_i)\}_{i=1}^N, \{F_{MoG}(y_j^*)\}_{j=1}^M | \mathbf{X}, \mathbf{X}^*) \prod_{i=1}^N f_{MoG}(y_i) \prod_{j=1}^M f_{MoG}(y_j^*). \tag{3.80}$$

Note these equations directly mirror the formulation in the case of univariate KCPs (Equations (3.6), (3.7) and (3.8) in Section 3.1). For the purpose of classification, I have found that the Gaussian copula and RBF kernel functions yields excellent classification accuracy, as illustrated by two non-trivial synthetic examples in the following section. More research is needed to determine if the performance of the KCP classifier can be improved by the choice of copula and kernel functions.

Just as in other classification problems, the design of the distance or *similarity* function often is the key to having good classification performance. The idea behind a distance or similarity measure is that data points that belong to the same class should have small distance or high similarity while data points in different classes should have large distance or similarity. Depending on the context of the classification problem, some distance measures may be more appropriate than others. For example, the Euclidean (L_2) distance is good for general use, the Manhattan distance (L_1) is good for reducing sensitivity to outliers or better reflect distances among locations within a city. Examples of other distance measures can be found [74].

The use of a distance function provides additional modularity within the kernel function. A common example is the RBF kernel function reproduced in Equation (3.81), where $d(\cdot,\cdot)$ denotes the distance function. A different distance function can be used without affecting the overall model framework. In fact, the features do not even have to be all of the same data type (i.e. they can be a mix of real, integer, ordinal, etc., as long as a distance measure is produced in the real space for the use of the kernel function

$$k_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2h} d\left(\mathbf{x}_i, \mathbf{x}_j\right)\right). \tag{3.81}$$

Once an appropriate distance measure is selected, the learning and inference procedures are exactly the same as the univariate KCP regression case. The likelihood function is

$$\mathcal{L}(\theta) = \mathbb{P}(\mathbf{y}|\mathbf{x}, \theta) = \mathfrak{c}(\{F_{MoG}(y_i)\}_{i=1}^{N}) \prod_{i=1}^{N} f_{MoG}(y_i). \tag{3.82}$$

Assuming the copula is Gaussian, the negative log-likelihood function is

$$-\log(\mathcal{L}(\theta)) = \frac{1}{2}\log|\Lambda| + \frac{1}{2}\mathbf{z}^T\Lambda^{-1}\mathbf{z} - \frac{1}{2}\mathbf{z}^T\mathbf{z} - \sum_{i=1}^{N}\log(f_{MoG}(y_i)), \tag{3.83}$$

where $\mathbf{z}^T = \{z_1 = \Phi^{-1}(F_{MoG}(y_1)), \ldots, z_N = \Phi^{-1}(F_{MoG}(y_N))\}$ is the transformed random vector $z_i \sim \mathcal{N}(0,1)$ and θ are the hyper-parameters for the KCP model (which includes the parameters of the mixture of Gaussian distribution and the kernel function). The likelihood function can be maximized by any gradient method such as the conjugate gradient algorithm. To classify new data points into the corresponding classes, the posterior distribution is first computed:

$$\begin{aligned}\mathbb{P}(\mathbf{y}^*|\mathbf{y},\mathbf{X},\mathbf{X}^*) &= \frac{\mathbb{P}(\mathbf{y},\mathbf{y}^*|\mathbf{X},\mathbf{X}^*)}{\mathbb{P}(\mathbf{y}|\mathbf{X})} \\ &= \frac{\mathfrak{c}(\{F_{MoG}(y_i)\}_{i=1}^N, \{F_{MoG}(y_j^*)\}_{j=1}^M|\mathbf{X},\mathbf{X}^*)}{\mathfrak{c}(\{F_{MoG}(y_i)\}_{i=1}^N|\mathbf{X},\mathbf{X}^*)}\prod_{i=1}^{M} f_{MoG}(y_i^*). \end{aligned}\tag{3.84}$$

As always, the classification decision using the KCPs can be made using the following maximum posterior criteria:

$$\underset{\mathcal{C}_k}{\text{argmax}} \quad \mathbb{P}(y = \mathcal{C}_k|\mathbf{X}^*, \mathcal{D}). \tag{3.85}$$

However, this is only an approximation due to the finite variance of the marginal distribution used (see Figure 3.11). Strictly speaking, the probability density around the class labels should be integrated according to the criteria

$$\underset{\mathcal{C}_k}{\text{argmax}} \quad \int_{-\xi}^{+\xi}\mathbb{P}(y = (\mathcal{C}_k + \delta)|\mathbf{X}^*, \mathcal{D})d\delta, \tag{3.86}$$

before comparison among classes is made, instead of Equation (3.85). Nevertheless, the approximation as provided in Equation (3.85) is typically sufficient so there is no need to actually perform the numerical integration in Equation (3.86).

Extensions to multi-class classification is trivial. The only difference is that the num-

ber of mixtures in the mixture model must be increased accordingly:

$$F_{MoG}(y) = \sum_{k=1}^{N_c} p_k \cdot \Phi(y; k, \xi), \tag{3.87}$$

$$f_{MoG}(y) = \sum_{k=1}^{N_c} p_k \cdot \phi(y; k, \xi). \tag{3.88}$$

Note that the sum of all p_k's is constrained to unity. The numerical value assigned to represent each class label is again completely arbitrary, as long as ξ is sufficiently small such that the individual Gaussian distribution do not overlap each other.

Clearly, classification is another area where the flexibility provided by the copula formulation employed by the KCPs can be appreciated. Unlike, GPC (as reviewed in Section 2.6.3) where the marginal distribution must be Gaussian, the use of copula functions in KCP classification allows a multi-modal marginal distribution (or any other arbitrary continuous distributions) to be used to directly mimic a multi-nominal distribution. The step of learning a latent process and subsequently integrating it out is completely eliminated. The complicated and computational intensive procedures of expectation propagation and Laplacian approximation can be avoided altogether. In fact, the procedure of KCP classification is almost exactly the same as KCP regression or forecasting. The only difference is that a multi-modal distribution is used for KCP classification.

3.9.1 Synthetic Example

In this section, a two dimensional synthetic example will be used to illustrate the power of a KCP classifier in solving a binary classification problem. To make this problem a little more challenging, the two classes are made *linearly inseparable*. That is, the two classes cannot be separated by a linear hyperplane. As such, conventional linear models will not perform well. The synthetic data set is illustrated in Figure 3.12a. The inner cluster is generated by a Gaussian distribution centered at the origin with a relatively small variance. The outer ring cluster is generated by a Gaussian distribution concentric with the former one but with a larger variance. The data points within a radius of 3

from the origin are removed so that there is no excessive interference. In this case, the only feature available for classification is the location of each data point $X = (X_1, X_2)$.

A simple KCP classifier is used to tackle this problem and consists of a Gaussian copula, a RBF kernel function, and the dual-Gaussian marginal distribution. MLE is used to learn the model paramters, and the unconditional joint and posterior distributions of the classifier are computed according to Equation (3.80) and (3.84). The posterior distribution of class $\mathcal{C}_0$, $\mathbb{P}(y = \mathcal{C}_0|\mathbf{X}^*, \mathcal{D})$ is shown in Figure 3.12b, where $\mathcal{D}$ denotes the set of training data set, i.e. label and feature pairs $\{\mathbf{y}, \mathbf{X}\}$, whereas $\mathbf{X}^*$ is taken to be the test set, consisting of the grid $\{-8, -7.9, \ldots, +7.9, +8\} \times \{-8, -7.9, \ldots, +7.9, +8\}$. Note that the posterior distribution predicts the general clustering of the two classes very well by visually comparing to the training data in Figure 3.12a.

It is interesting to note that $\mathbf{x}^* = (8, 8)$, is sufficiently far away from all the training points that the posterior distribution $\mathbb{P}(y^*|\mathbf{x}^*, \mathcal{D})$ approaches to the unconditional marginal distribution as shown in Figure 3.12c. In fact, the rate of relaxation to the unconditional marginal is exponential due to the exponentiation in the RBF kernel function. To understand this, first consider the joint distribution of a pair of training and test points $\mathbb{P}(y^*, \mathbf{y}|\mathbf{x}^*, \mathbf{X})$,

$$\mathbb{P}(y^*, \mathbf{y}) \to \mathbb{P}(y^*) \cdot \mathbb{P}(\mathbf{y}) \;\text{ as }\; d(\mathbf{x}^*, \mathbf{x}) \to +\infty \;\text{ and }\; k_{RBF}(\mathbf{x}^*, \mathbf{x}) \to 0. \tag{3.89}$$

Thus the posterior distribution $\mathbb{P}(y^*|\mathbf{x}^*, \mathcal{D}) \to \frac{\mathbb{P}(y^*)\cdot\mathbb{P}(\mathbf{y})}{\mathbb{P}(\mathbf{y})} = f_{MoG}(y)$, yielding the unconditional marginal distribution.

In this example, $p_{MLE} = 0.663$, roughly corresponding to the fraction of data points in class $\mathcal{C}_0$ ($\sim 62.3\%$). In such case, the class assignment decision simply reduces to picking the class with the largest weight in the multi-Gaussian distribution (which was learned from the training data),

$$\underset{\mathcal{C}_k}{\text{argmax}} \;\; \mathbb{P}(y = \mathcal{C}_k|\mathbf{X}^*, \mathcal{D}) \;\; = \underset{\mathcal{C}_k}{\text{argmax}} \;\; p_k. \tag{3.90}$$

Even more interesting aspects of the KCP classification model can be illustrated by

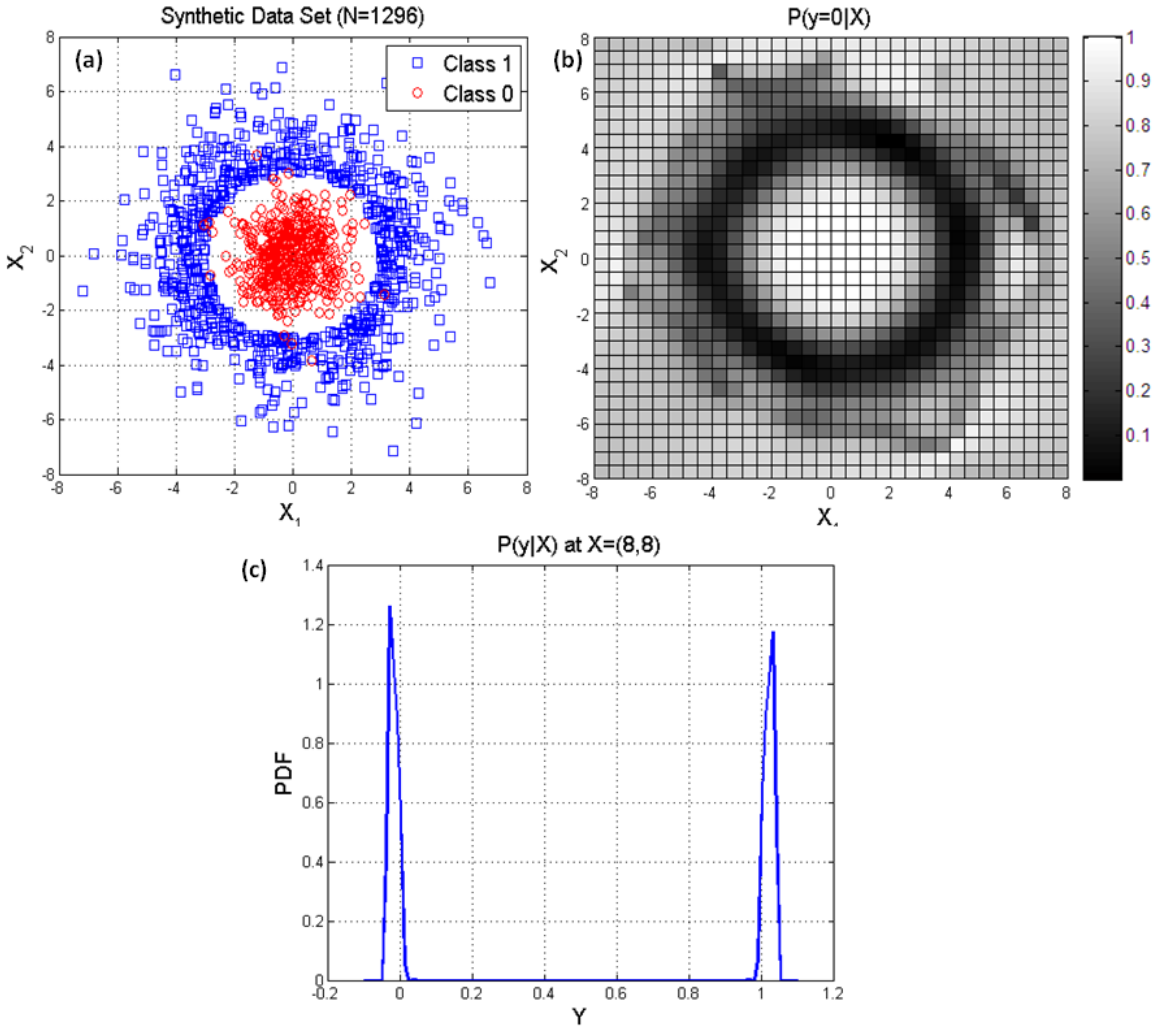

Figure 3.12: Synthetic example for KCP classification. (a) A two-dimensional synthetic data set for illustrating the power of *binary* classification with KCPs. Note, this data set is challenging for conventional linear models because it is not linearly separable. (b) The posterior probability map of class 0, $\mathbb{P}(y^* = \mathcal{C}_0|\mathbf{X}^*, \mathcal{D})$. (c) The posterior probability at $\mathbf{x}^* = (8, 8)$.

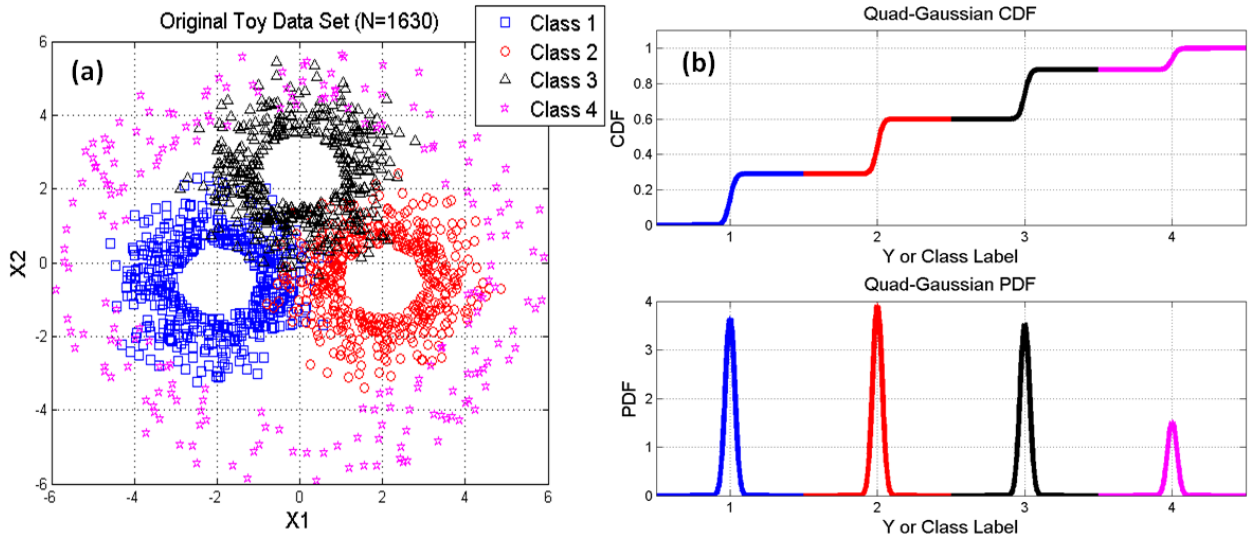

Figure 3.13: Synthetic example for multi-class classification. (a) The synthetic data set consists of four classes: three tight rings are generated by a Gaussian distribution with a standard deviation of 1 with the center lobe (radius of 1) removed; while the bigger ring is generated by a Gaussian distribution with a standard deviation of 5 with the center lobe (radius of 4) removed. There are total number of 1630 data points, distributed among the 4 classes in this proportion: $\{29.3\%, 29.2\%, 29.7\%, 11.8\%\}$. (b) Quad-Gaussian marginal distribution with MLE parameters $p_{MLE} = \{0.29, 0.31, 0.28, 0.12\}$.

considering a multi-class classification example. A 4-class synthetic data set is shown in Figure 3.13a. The four clusters are partially overlapped at different regions. For this classification task, a KCP classification model consisting of a quad-Gaussian marginal (as prescribed in Equation (3.87) and (3.88)), and a Gaussian copula function with an RBF kernel function is used.

The data set in Figure 3.13a is used as the training set for the KCP classifier using maximum likelihood. The unconditional quad-Gaussian marginal distribution with maximum likelihood parameter is shown in Figure 3.13b. The relative heights of the peaks approximately equal to the relative frequency of each class in the training set as expected.

Again the posterior distribution $\mathbb{P}(y^*|\mathbf{X}^*, \mathcal{D})$ is used to make classification decisions across the input feature space. Figure 3.14 depicts examples of the posterior distributions $\mathbb{P}(y^*|\mathbf{X}^*, \mathcal{D})$ at different test points $\mathbf{X}^*$ in the input feature space. A consistent color code is used throughout for ease of visualization. To construct a map of classification results, a grid $(X_1, X_2) = \{-6, -5.9, \ldots, 5.9, 6\} \times \{-6, -5.9, \ldots, 5.9, 6\}$ is used as the test set. The posterior distributions at each point are computed. To arrive to the final classification decision, the maximum posterior criterion in Equation (3.85) is used and is shown in 3.15a. The maps of posterior distributions at the class labels $\mathbb{P}(y^* = \mathcal{C}_i|\mathbf{X}^*)$ for $i = \{1, 2, 3, 4\}$ are also shown in 3.15b. Note that there is a thin ring of $\mathcal{C}_3$ (Class 3) extends around $\mathcal{C}_1$ and $\mathcal{C}_2$. One possible explanation is that, at the location of the ring, training points are only sparsely populated, and given the exponential decay of similarly prescribed by the RBF kernel function, the posterior distribution favors the class with the highest relative frequency in the training set which is Class $\mathcal{C}_3$.

In general, like the prediction problem, classification is a difficult problem and faces a long list of unique challenges. To obtain superior performance, models must take advantage of any domain knowledge and fine structure in the training data. This section merely provides a preview of the capability of the KCP model in this area. More research is required to take full advantage of the KCPs. Many questions are still unanswered. For example, what is the best way to incorporate domain knowledge in the KCP model? Can the training data be a guide for selecting the copula and kernel functions? Is there another more suitable marginal distribution?

Regrettably, these topics fall outside the scope of this work. The detour presented in this section is intended to show the versatility of the KCP model. The remaining chapters will refocus back on the application of the KCPs in time-series analysis.

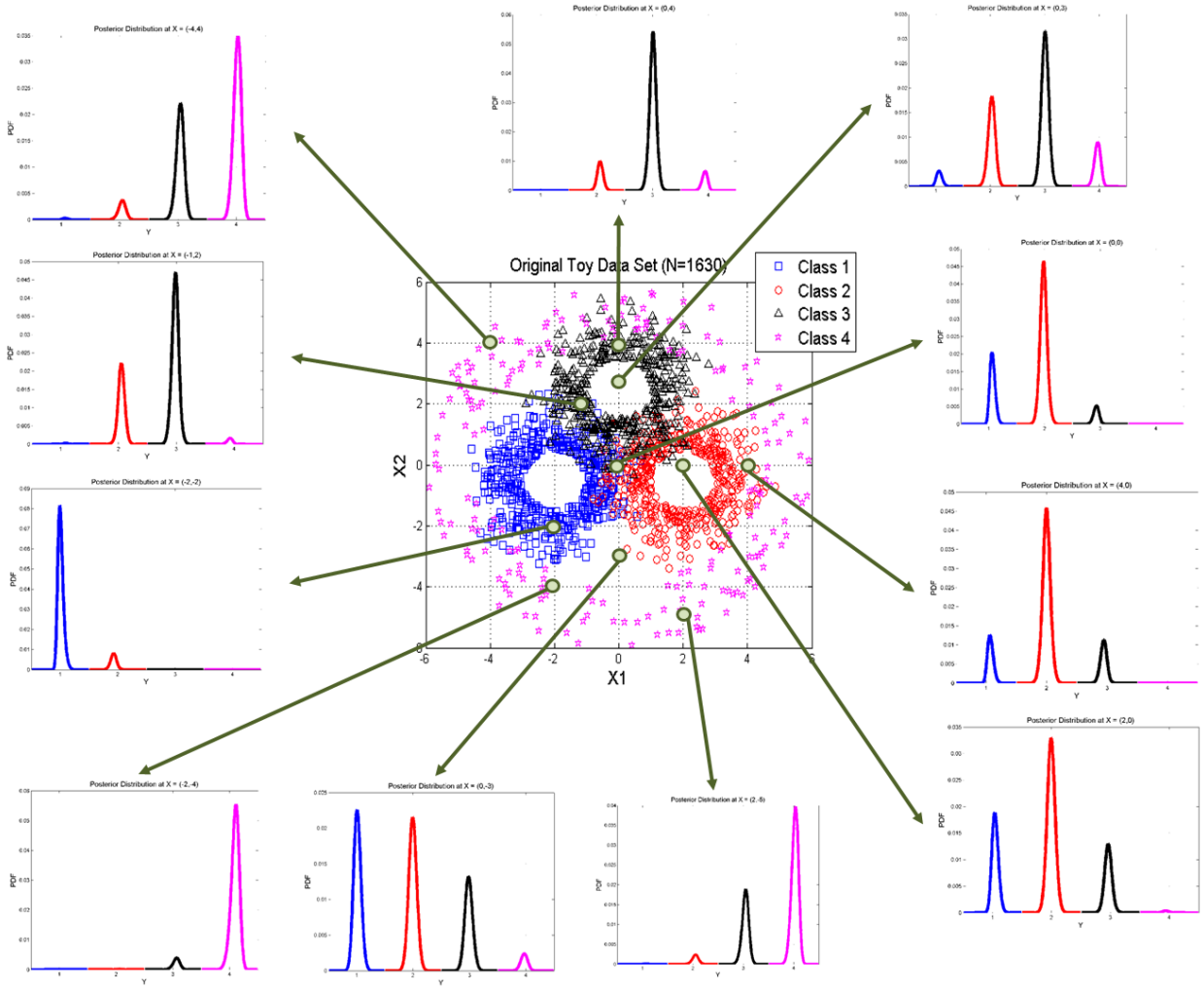

Figure 3.14: A map of posterior distributions at different points in the input feature space. The probability densities of different portion of the posterior distributions are color coded to the corresponding class label for ease of visualization.

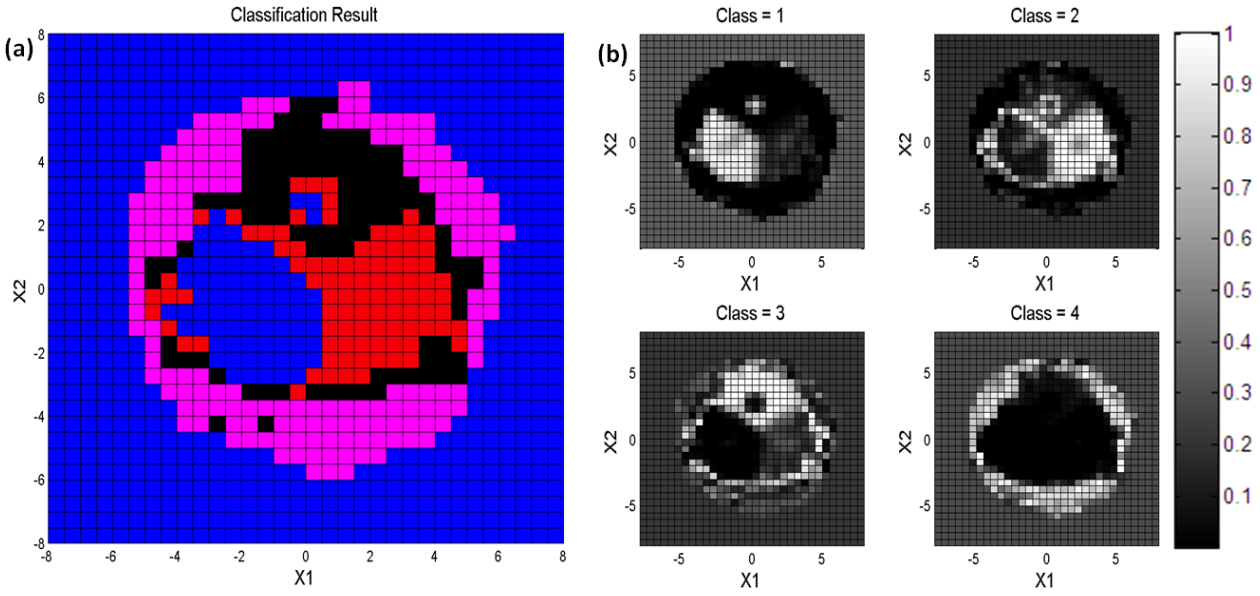

Figure 3.15: Classification result of the synthetic data set using the KCP classification model. In panel (a) the classification result based on the maximum posterior criterion in Eqn. 3.85 is shown. The maps of posterior distributions at the class labels $\mathbb{P}(y^* = \mathcal{C}_i|\mathbf{X}^*)$ for $i = \{1, 2, 3, 4\}$ are shown in panel (b).

Chapter 4

Applications

The KCP model is a versatile tool in time-series analysis and regression analysis. To demonstrate this point, this chapter showcases the KCP model in action with a few real-life applications. The selection of applications, biased by my personal interests, will be mainly focused on financial time-series. For such applications, conventional models tend to perform poorly, thus the need for innovative techniques is urgent and very tangible. Furthermore, a non-financial application in temperature prediction will be demonstrated in this section to provide a broader scope of applicability.

Before diving into the specific applications, however, the overall design methodology with the KCP model is presented in Section 4.1. This serves as a design guideline for the real-life applications to be presented in Section 4.2.

4.1 Overall Design Methodology

The proposed design methodology with KCPs can be succinctly summarized in the template depicted in Figure 4.1.

In general, when applying the KCP model, first one performs some data pre-processing such as detrending, removal of seasonality, standardization, and for financial time-series, deciding whether to work with price levels or returns. Further details are discussed in

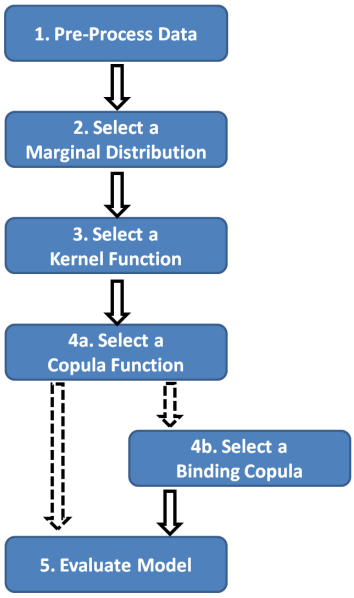

Figure 4.1: KCP design methodology.

Section 4.1.1. Then the specific forms of the major components of a KCP (i.e. marginal distribution, kernel function, and copula function) must be selected. The details of the selection processes are thoroughly discussed in Section 4.1.2, 4.1.3, and 4.1.4 respectively. Finally, the performance of the model on the data set under consideration is evaluated. The details of the application of various performance evaluation techniques and methodologies described in Sections 3.7 and 4.2.3 is given in Section 4.2.6. For multivariate data sets, one must also select a binding copula function. The details of which will be discussed in Section 4.2.7.

4.1.1 Data Pre-processing

Real-life data comes in all forms, with different structures and possibly with distortions embedded in them. However, some of these structures are simply distortions or well-known oddities in the data which obstruct the objective of the data analysis. Thus the data must be put into the right form, and known structures and distortions must be removed before the data is fed to the model of choice.

The simplest and most common forms of data pre-processing include standardization, and removing deterministic trends (e.g. linear growth in consumer prices) and seasonality components (e.g. seasonal changes in natural gas prices). Both of these operations allow one to focus better on the stochastic component of the data. For instance, standardization removes long-term averages and normalizes the variance. Similarly detrending and deseasonalization removes distracting deterministic changes in the data.

For some applications, however, specific pre-processing is needed. The decision of whether or how the pre-processing is performed could significantly impact the final results, and domain-specific knowledge is often required. In econometrics, for instance, one common decision practitioners face is whether to work with the levels (denoted P_t) or values at which the data is provided (e.g. prices of a commodity) versus the logarithmic changes over an interval (e.g. daily returns of a commodity, denoted $r_t = \ln(P_t) - \ln(P_{t-1})$). Stationary processes are usually favored in these situations, and the stationarity can be determined by the unit-root test.

Another example of pre-processing can be found when handling futures[1] prices. To construct an extended continuous time-series from futures contracts that come and go out of existence, practitioners must decide when to *roll-over* the contracts (i.e. to replace the price series of one contract with the next one) and whether to introduce level shifts to render the composite time series *continuous*[2].

The bottom-line is that the type of data pre-processing must be chosen on a case-by-case basis and depends on the type of data being analyzed. However, it is important that the process is deterministic so that it does not introduce spurious stochastic features and it can be easily reversed at the end of the data analysis process.

[1] A futures contract is a standardized covenant under which two parties agree to exchange an underlying asset (e.g. commodities such as coffee beans, pork bellies, etc.) for an agreed price at a future date. On that future date (aka the contract expiry date), the physical goods are delivered and money changes hands.

[2] This is a difficult issue to define for financial prices, especially for commodities with volatile prices since price jumps often occur within the life time of a single contract.

4.1.2 Marginal Distribution Selection

As illustrated in Section 3.1.3, having the appropriate marginal distribution in the KCP specification is crucial. In principle, the KCP models impose no restriction on the choice of marginal distribution. Thus any parametric marginal distribution can be used, even an empirical one. Ultimately, the data must be the guide for such selection.

To this end, I recommend the following procedure. Treat each sample in the time-series is as *iid* samples from a target distribution. Start with all possible parametric families of distribution, then the closest parametric family is chosen with the help of AIC or BIC, and the associated parameters are learned with MLE. In practice, one can also narrow down the choices of parametric families by visually inspecting the histogram of the data series.

In cases where no parametric distribution performs reasonably well, the empirical non-parametric marginal distribution could be used as well.

A word of caution, the proposed method is not strictly correct as it assumes that the outcomes are all *iid*, i.e. the random process is being treated as a single random variable with values. As we know, a random process is a collection of random variables. Thus a sample path of a time-series (which is a random process), represents a series of draws from potentially different distributions. Therefore, collapsing all the samples from a time-series to form a histogram or *iid* data set is axiomatically incorrect. Nevertheless, as will be shown later in this chapter, this method turns out to be simple and effective in choosing the class of marginal distribution (but not the specific parameters themselves). A loose theoretical justification would be its close resemblance to the strong independence assumption in the *naïve Bayes* probabilistic model [7].

4.1.3 Kernel Function Design

As mentioned in earlier chapters, the kernel function is one of the most important ingredients of the KCP model. It allows behaviour such as mean-reversion and periodicity to be captured in a compact manner. For more complex behaviour, new kernels can be created by combining multiple kernel functions through addition, multiplication, convolution, or tensor product. Much work in this area has been done in the literature of Gaussian Processes [85] and kernel methods [94]. As shown in Section 3.3, a customized kernel function can also be created if the SDEs describing the desired process is known. When dealing with real data, however, the auto-correlation functions (ACFs) of the first few moments[3] can often give us important hints for guiding the design or the selection of the kernel function.

For example, in some of the temperature time series data set to be discussed in more details later in this chapter, observing the ACFs of the first four moments gave us important hints. First, from the ACF of the second moment, we learned that the variance of the time-series has a semi-annual periodicity. Further, upon examining the ACF of the first moment, we observed substantial auto-correlation for lags less than 10 days. These two facts together guided us in making the decision to design a customized kernel function using a mean-reverting SDE with a periodic variance process as shown in Section (3.3.2).

In other cases, spectral analysis (such as Fourier transform) can also help to reveal multiple periodicities that may exist in a time-series, and a sum of periodic kernel functions (Equation (3.15)) can be used.

[3]Specifically, in this work, the *k-th moments* refers to the *k-th* power of the original time series, assuming the original time series has been standardized. For instance, the second moment of the time series $\{y_1, y_2, \ldots, y_n\}$ is given by $\{y_1^2, y_2^2, \ldots, y_n^2\}$.

4.1.4 Copula Functions Selection

For univariate KCP, the main purpose and contribution of the copula function is the separation of the joint dependency and the marginal behavior. This separation allows the flexible modeling of non-Gaussian marginal distributions while keeping learning and inference tractable. Further, a natural time-ordering of data can be introduced through a proper choice of copula functions. Specifically, the *elliptical class* copula function[4] allows a kernel function to be embedded in it such that a sense of distance, and thus time-ordering, can be introduced. Without such time- or spatio-ordering, the KCP would not satisfy the definition of a stochastic process or random-field. The elliptical class contains many parametric families, such as the family of Pearson distributions, which in turn contains the Gaussian, Student's t, Cauchy, and Laurentian distributions. In this work, it is found that while copula functions do help to capture higher order dependencies across time, the exact choice of copula functions for the univariate KCP provides only incremental improvements on the final performance. Thus it is recommended that the simplest elliptical copula (computationally or otherwise) such as the Gaussian or the Student's t-copula be used. If computational resources or the amount of data is not a constraint, one can use the most general class of copula function and allow the appropriate parameters to be learned. Alternatively, multiple copula functions can be used for consideration and a model selection criteria (see Section 3.7) applied to select the best model.

[4] An elliptical copula function typically has the following form:

$$\mathbb{C}(u_1, \ldots, u_n) = F_\Sigma(F_1^{-1}(u_1), \ldots, F_n^{-1}(u_n))$$

where $F_i^{-1}(u_i)$ are standardized univariate elliptical distributions of the same type as F_Σ. For more details, please refer to A.5.1.

4.2 Real-Life Applications

In addition to being a powerful and flexible tool for general regression analysis, we will show that the KCP model is also useful in various applications. A wide range of time-series data are chosen in this section to illustrate the versatility and modeling power of the KCP model. These include the time-series of the volatility index (VIX), prices of crude oil futures, foreign exchange rates, and daily temperatures recorded in the United States. More details of each data are given in the respective sections below.

4.2.1 Competing Models and Overall Data Modeling Strategy

In this section, the details of the overall data modeling strategy pertaining to the specific data is outlined. The specifics of the data analysis and results of each data set are detailed in later sections.

In this chapter, six competing models, as listed in Table 4.1 will be applied to a number of real-life data sets to evaluate their relative performances in terms of modeling or predictive power, computational efficiency, and model complexity. In particular, the major rivals of the KCP model are included, namely the GARCH(1,1), warped GPs, and the GPs.

The GARCH(1,1) model is included here because it is one of the best-known nonlinear models for heteroskedastic processes [9][50]. Further, its conceptual and computational simplicity and its versatility has made it one of the most frequently used time-series analysis tool. The t-innovations are used instead of the typical Gaussian-innovations to allow greater flexibility to better adapt to the financial time-series considered in this chapter.

The GP is a natural basis for comparison because it can be seen as a specific instantiation of the KCP framework. Further, due to its popularity in the research community, it is important for us to point out the trade-offs between the need for modeling non-

Table 4.1: A list of competing models is tabulated along with the number of parameters involved. The number of parameters is important for model selection based on AIC and BIC (See Section 3.7 and Section 4.2.6.)

	Competing Models	No. of Parameters	
		OU Kernel	Hetero. Kernel
1.	GARCH(1,1) model with t-innovations	4	
2.	warped GPs with neural-nets style warped function	9	12
3.	GPs	3	6
4a.	the Gaussian copula with skew-Student's t marginal (GCStM) KCP	5	8
4b.	the Gaussian copula with skew-normal marginal (GCSM) KCP	4	7
5.	Student's t-copula with Gaussian marginal (TCGM) KCP	5	8
6a.	Student's t-copula with skew-t marginal (TCStM) KCP	6	9
6b.	Student's t-copula with skew-normal marginal (TCSM) KCP	5	8

Gaussian data versus computational requirements. Trade-offs are implicitly made by researchers everytime the generic GP is used for data analysis without understanding its limitation.

The warped GP is chosen here as it is one of the popular extensions of the GPs to handle non-Gaussian and non-linear data. The neural-net style (or sum of hyperbolic tangents, Equation (2.13)) warping function is used throughout. The number of hyperbolic tangents is chosen and fixed to be two since it roughly represents the optimum point of trade-off between modeling power and model complexity [5].

Finally, multiple KCPs with different configurations (i.e. combinations of marginal distributions and copula functions) are chosen for further comparison. The different combinations are listed in Table 4.1. These configurations are chosen to compare and contrast the relative importance of the marginal distributions and copula functions in different applications.

To conduct fair comparisons, the GPs, warped GPs, and KCPs above always use the same kernel function for a given application. The OU-kernel function (Equation (3.5)) or heteroskedastic kernel function (Equation (3.15)) are selected according to the characteristics exhibited by the data.

In multi-series analysis, the Gaussian- and Student's t-copula are used as a binding copula to describe the interdependencies among the currency rates and temperature series.

4.2.2 Model Training Method

The main goal of this investigation is to evaluate the predictive power of each model. To this end, for a given time-series, a sliding window of 100 days is used for MLE training and then a 1-day-ahead prediction is made at every step. This window size is chosen to

[5] Modeling selection using BIC was used to select the number of hyperbolic tangents needed for the WTI, VIX, and FX data sets

be sufficiently big (i.e. much bigger than the learned length-scale parameters) so that the conditions outlined in Section 3.5.1 are satisfied and the approximation in Equation (3.60) holds. Thus the data points in the history beyond the window size can be safely ignored. Further, the same training window size is used for all models deliberately to facilitate fair comparison. Alternately, an optimal window size can be chosen for each model such that each model is performing at its best; or a *cross-validation* scheme can be used for model learning[6].

The ML parameters from the current window is used to initialize the search of the next window. This should serve as a good starting point for the optimization. The first set of parameters was learned with 20 random-restarts[7]. The range of random values for each parameter is selected with the help of the data examining procedure as discussed in Section 4.2.5. The set of 1-day-ahead predictions produced allows the performance of the models to be ranked on a predictive basis as further elaborated below.

4.2.3 Model Selection and Performance Evaluation

In this chapter, the competing models will be ranked based on the *in-sample* and *predictive* model selection metrics introduced in Section 3.7.

Specifically, every step in the sliding training window mentioned in the previous section produces a likelihood, AIC, and BIC value. The median of those values are used for ranking purposes, such as the ones shown in Figure 4.2. The 1-standard deviation error bars are also shown to provide an idea of the spread of those measures. Furthermore, the predictive powers of each model are evaluated by the use of the probability integral

[6]In cross-validation, the available data set is typically divided in to M segments, learning is done on $M-1$ segments, and performance is evaluated on the remaining data segment. This exercise is repeated M for all permutations of segments and the overall model performance is obtained by averaging across the M trials

[7]For each random restart, the initial values of the parameters are randomly chosen, and a gradient method (such as the conjugate gradient method) is used to maximize the likelihood function. The process is repeated 20 times, the final set of parameters is chosen to be the set that resulted in the highest likelihood value.

transform (PIT).

Finally, sample paths are generated from each model using the learned parameters as a sanity check to verify if the models have in fact learned something sensible.

4.2.4 Summary of Data Sets

Volatility Index (VIX)

The volatility index (VIX), also commonly known as the *fear-gauge* of the stock market, is a measure of the implied volatility of the options of stocks listed the S&P 500. It is an index for estimating the implied volatility[8] of the S&P 500 index over the next 30 days.

Simplistically, it is computed by averaging the implied volatilities of the short-term options. More specifically, it uses a kernel-smoothed estimator that takes the current market prices for all out-of-the-money call and put options for the front and second moth expirations. The VIX is calculated and disseminated in real-time by the Chicago Board Options Exchange (CBOE)[15].

The VIX data set under consideration here is the daily closing level of the index between January 3, 2006 to April 16, 2008, a total of 575 data points.

West Texas Intermediate (WTI) Crude Oil Futures

Crude oil is composed of long hydrocarbons and can be refined to other fuels such as gasoline, heating oil and other raw materials such as plastics. Crude oil varies in quality, and thus price. It is typically classified by the geographic location where it is produced, its density, and sulfur content. For example, the West Texas Intermediate (WTI) is a type of light sweet crude, which refers to its relatively light density and low sulfur

[8]The implied volatility is the only free parameter of the Black-Scholes option pricing model [50]. As such, this parameter is determined by the supply and demand of the market and is synonymous with the price of the option itself. Unlike historic volatility, which is a measure of variability of prices in the recent past, the implied volatility indicates the market's current perception of future levels of variability a certain index or security will experience as implied by the current observable conditions.

content. The geographic location is important because it affects the transportation costs to refineries.

The WTI crude is also used as a benchmark in oil prices. The associated futures contracts is traded in the New York Mercantile Exchange (NYMEX). Each contract controls 1,000 barrels of crude. Contracts with expirations ranging from less than 30 days to 9 years are available for trading. All contracts are priced in US dollars on a per barrel basis.

The WTI data set considered here is the daily closing prices of the front month (i.e. nearest-expiry) contracts from January 2, 1992 to December 29, 1995, totaling 1004 data points.

Foreign Exchange Rates

Foreign exchange rates is a subject of interest for a wide variety of market participants and housewives[9] alike: travelers try to get favorable rates for their next trip, corporations must manage their currency exposures carefully to meet their obligations to suppliers and pay-rolls, central banks must monitor exchange rate movements closely to ensure their country remains globally competitive, not to mention the flocks of currency traders around the world watching the every move of every currency.

The floating fiat currency system that is currently in-place around the world means currency values are only set relative to each other. Thus it is expected that a rich co-dependent structure embedded will be in the currency rates data. In this chapter, we will examine the exchange rates of five developed countries in relation to the US dollar (USD). These currencies include the Australian dollar (AUD), the Canadian dollar (CAD), the

[9]It has been said that the majority of foreign exchange activities in Japan are driven by individual investors who are risk-adverse and yield-hungry. Those individual investors routinely sell Japanese Yen and deposit their money in other high-yielding currencies such as the Australian dollar and New Zealand dollar. Since family finances are typically handled by the female head of the household, the individual currency investors in Japan are dubbed *Mrs. Watanabe*. It is also one of the most common last name in Japan and aptly carries the meaning of *cross-border*.

Euro (EUR), the Great British Pound (GBP), and the Japanese Yen (YEN). For each time-series, the daily values of the exchange rates between January 2, 2002, and December 31, 2005, totaling 1095 data points, are available. Specifically, the data was taken three years after the introduction of the Euro dollar, thus any shock it may have introduced to the world currency market would have subsided. This data set is freely available from OANDA [18].

Temperature Data

To showcase the modeling power of KCPs with multiple time-series, we consider the daily maximum temperatures recorded by the United States Historical Climatology Network (HCN) [55]. Data from as far back as the early 1800s from a network of 1219 weather stations are freely downloadable from the network's website. This study will consider the data from a few arbitrary three-year periods from 156 stations of various states. Missing data is commonplace in such data, as equipment is moved and maintained; however, KCPs handle missing data naturally.

The latitude/longtitude and elevation coordinates for each weather station are also known. We will later use this information to help learn the dependency structure.

4.2.5 Data Inspection and Pre-processing

Data inspection and pre-processing is the first step in data analysis, as mentioned in Section 4.1, where one decides what type of pre-processing is needed. In the KCPs case, one must also decide on the type of marginal distribution and kernel function to be used.

First, consider the VIX data set as shown in Figure 4.2a. As seen, there are a lot of abrupt changes or jumps throughout. More specifically, there are periods of high rate of change or volatilities, and there are periods of calms where the index does not change much. This is known as *volatility clustering*, and it is a well known *stylized fact* of volatility series as markets go through periods of frenzy or panic, which then returns

to calmer trading. This volatility clustering phenomenon is also easily observable in Figure 4.2b, where the standardized logarithmic returns are shown. The logarithmic returns (instead of the index level itself) is the preferred quantity for analysis in our case as it transforms the time-series from a positive-only series to a double-sided series. The standardization of the data also helps to reduce the number of parameters to be estimated by two, namely the mean and variance parameters. Furthermore, it also serves as a simple but effective detrending mechanism. In fact, the same pre-processing is done to all subsequent data series, except for the temperature data. Then, upon examining the autocorrelation functions (ACFs) of the first four moments of the logarithmic returns in Figure 4.2c, virtually no systematic correlation structure is readily identifiable other than some incremental negative correlations in the first moment and some incremental positive correlations in the second moment given a lag of a couple of days. Thus the versatile OU kernel is used in this case. On the other hand, consider the empirical histogram of the logarithmic returns in Figure 4.2d. The data is high in kurtosis and possibly positively skewed. Clearly, a Gaussian distribution cannot account for excess kurtosis and skewness. Thus, a skew-t distribution will be used for the KCP models. Note that, this inspection of the histogram also allows us to use the best-fit parameters of the skew-t distribution as the initial values of the maximum likelihood estimation.

The situation is quite similar for the WTI series as shown in Figure 4.3. A notable difference is that there seems to be some mean-reversion of the crude oil prices during that period at around \$18 to \$19 per barrel. Mean-reversion is not uncommon for commodities prices, driven by factors in supply and demand. For example, a build-up in inventory may cause prices to drop, thus resulting in a temporary slow-down in production. Prices then return to a more reasonable level as excess inventory is consumed. If the production-cut was overdone, inventory will dwindle too much and prices will shoot up once demand recovers. Prices can fluctuate due to geo-political tensions and other factors as well. Given this mean-reverting behavior, the OU kernel again is the ideal choice since it

can be derived from a mean-reverting process as shown in Section 3.3.2. The skew-t distribution is again chosen as the marginal distribution for the KCP models because of the excess kurtosis.

The same pre-processing procedure is performed on the five currency rate series. The Japanese Yen rates (with respect to the US dollar) are shown in Figure 4.4. Similar to the previous two cases, logarithmic returns and the skew-t distribution are chosen. However, notice the second moment ACF of the logarithm return series, i.e. the ACF of the square of the data, exhibits a periodic structure of a period of roughly seven days. This is due to the lack of heavy trading activities on the weekends [10]. The heteroskedastic kernel was designed in Section 3.3.2 specifically for this purpose. To reduce the amount of learning, the period of the kernel can be fixed at seven days.

Finally, the same procedure was applied to the temperature series, as shown in Figure 4.5. A sinusoidal trend corresponding to the change of seasons is subtracted from the temperature levels in order to focus on the stochastic nature of the data. A customized sinusoidal function is used for each weather station based on the least-square error criteria. In this case, the detrended data (as shown in Figure 4.5 b) is then used directly without taking the logarithmic returns. The ACFs of this data set are very different from the financial series considered so far. The first moment ACF shows some strong short-term correlation which implies recent temperature trends tend to persist. That is, high temperature days tend to be followed by high temperature days and vice versa. The second moment ACF also experiences an annual cycle as illustrated in Fig. 4.5c. This provides an important hint in designing or selecting the type of kernel function that would be appropriate for the analysis. Again, the heteroskedastic kernel function was designed specifically to fulfill the above features. Lastly, note that without the excess kurtosis, the choice of a skew-normal distribution works well for this data series. This is subsequently confirmed by a *Chi*-square test. Again, the periodicity observed in the ACF and the best-fit parameters of the skew-normal distribution can be used as the initial values of

those parameters during MLE.

It is important to note that the above data inspection is not necessary if one intends to use the KCPs as an off-the-shelf tool. For example, practitioners use GPs with the RBF kernels or use GARCH(1,1) with Gaussian innovations by default if absolute prediction accuracy is not required.

On the other hand, the data inspection procedure recommended in this section presents a simple way to incorporate additional domain knowledge in guiding the model construction process. It provides initial values for the parameters for MLE and also a basis for forming the prior distribution for Bayesian learning. In fact, the initial values of all the marginal distribution parameters, such as the degree of freedom ν and skewness λ, can be obtained during the selection of the best marginal distribution as depicted in 4.5d. Similarly, the initial value of the length scale parameter κ in the kernel function (Equation (3.41)) can be estimated by $\hat{\kappa} = -\log(\rho|_{\tau=1})$, where $\rho(\tau)$ denotes the ACF for the first moment at lag τ; whereas the initial value of the periodicity parameter T of the variance process in the Heteroskedastic kernel (Equation (3.41)) can be learned from inspecting the second moment ACF directly. The ability to come up with these informative estimates for the parameters serves well to reduce the number of random-restarts during model learning and perhaps even lessen the need for Bayesian learning.

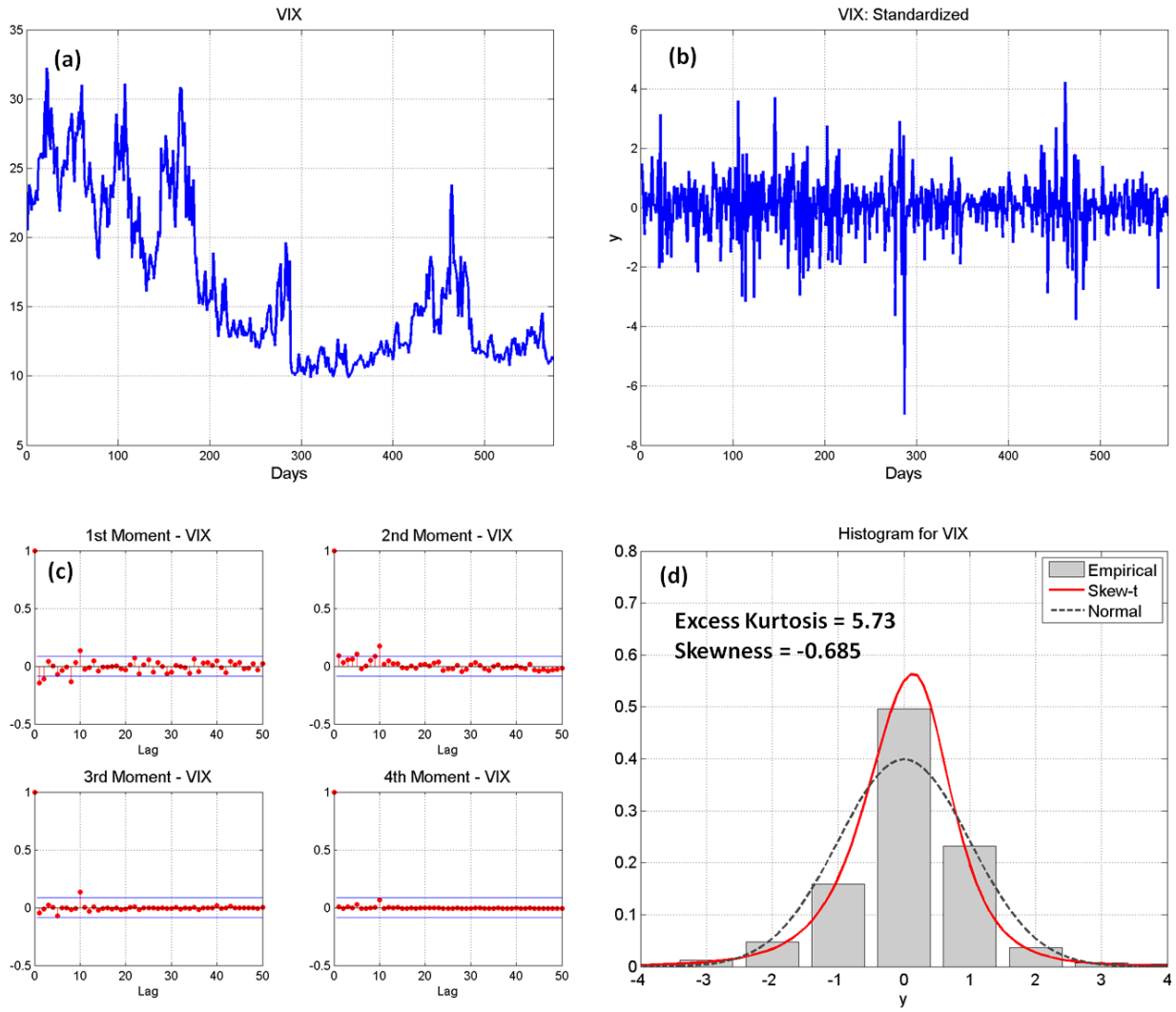

Figure 4.2: VIX: Data Examination: (a) The daily closing values of the VIX index between Jan 3, 2006 and April 16, 2008; a total of 575 data points. (b) The daily logarithmic returns of the VIX index. (c) The ACFs of the first four moments of the log-returns of VIX. (d) The histogram and best-fits with Gaussian and Skew-t distributions of the log-returns of VIX.

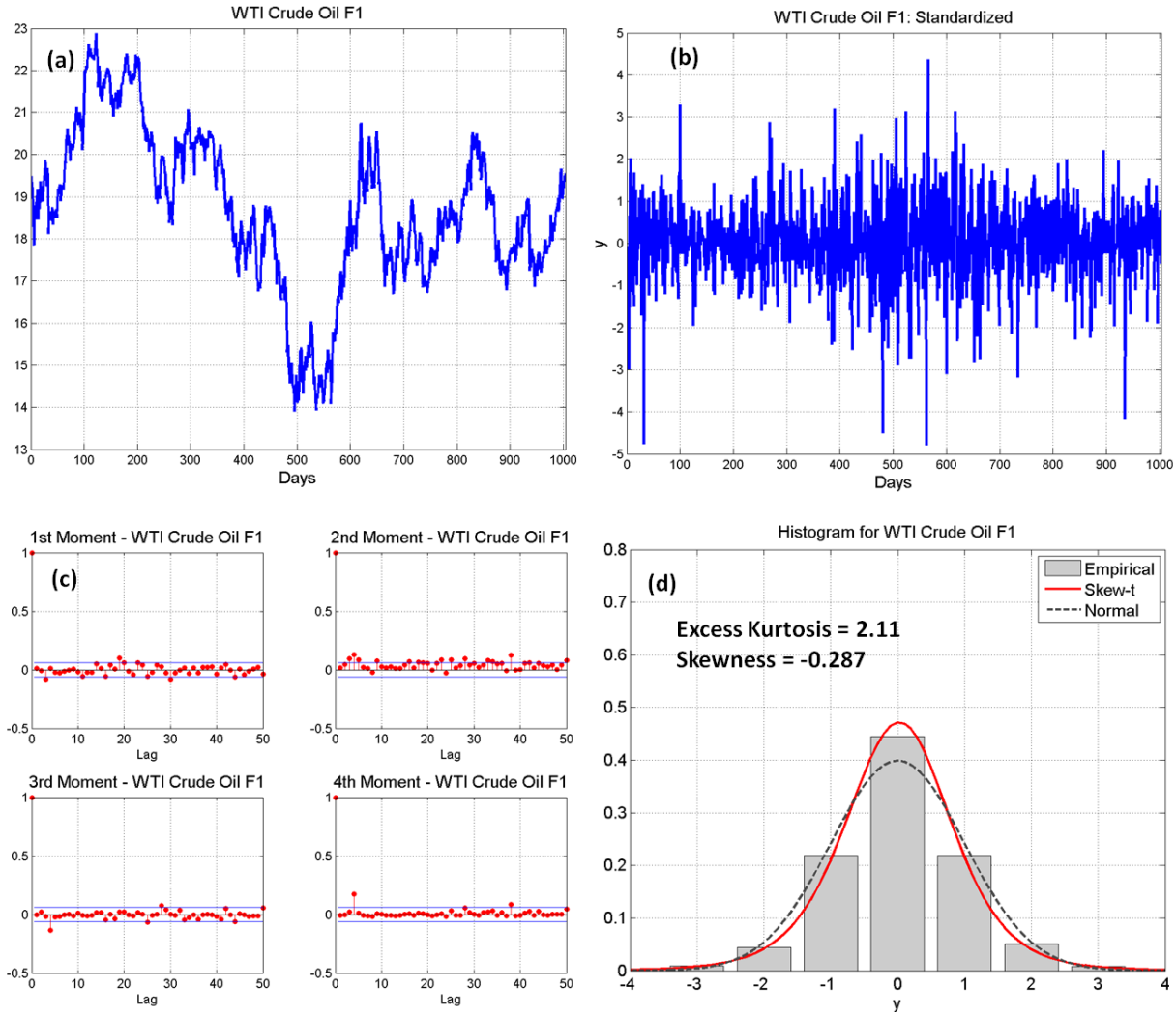

Figure 4.3: WTI: Data Examination: (a) The daily closing prices of the WTI Crude Oil Futures (front-contract) between Jan 2, 1992 and December 29, 1995; a total of 1004 data points. (b) The daily logarithmic returns of the contract prices. (c) The ACFs of the first four moments of the log-returns of the contract prices. (d) The histogram and best-fits with Gaussian and Skew-t distributions of the log-returns of the contract prices.

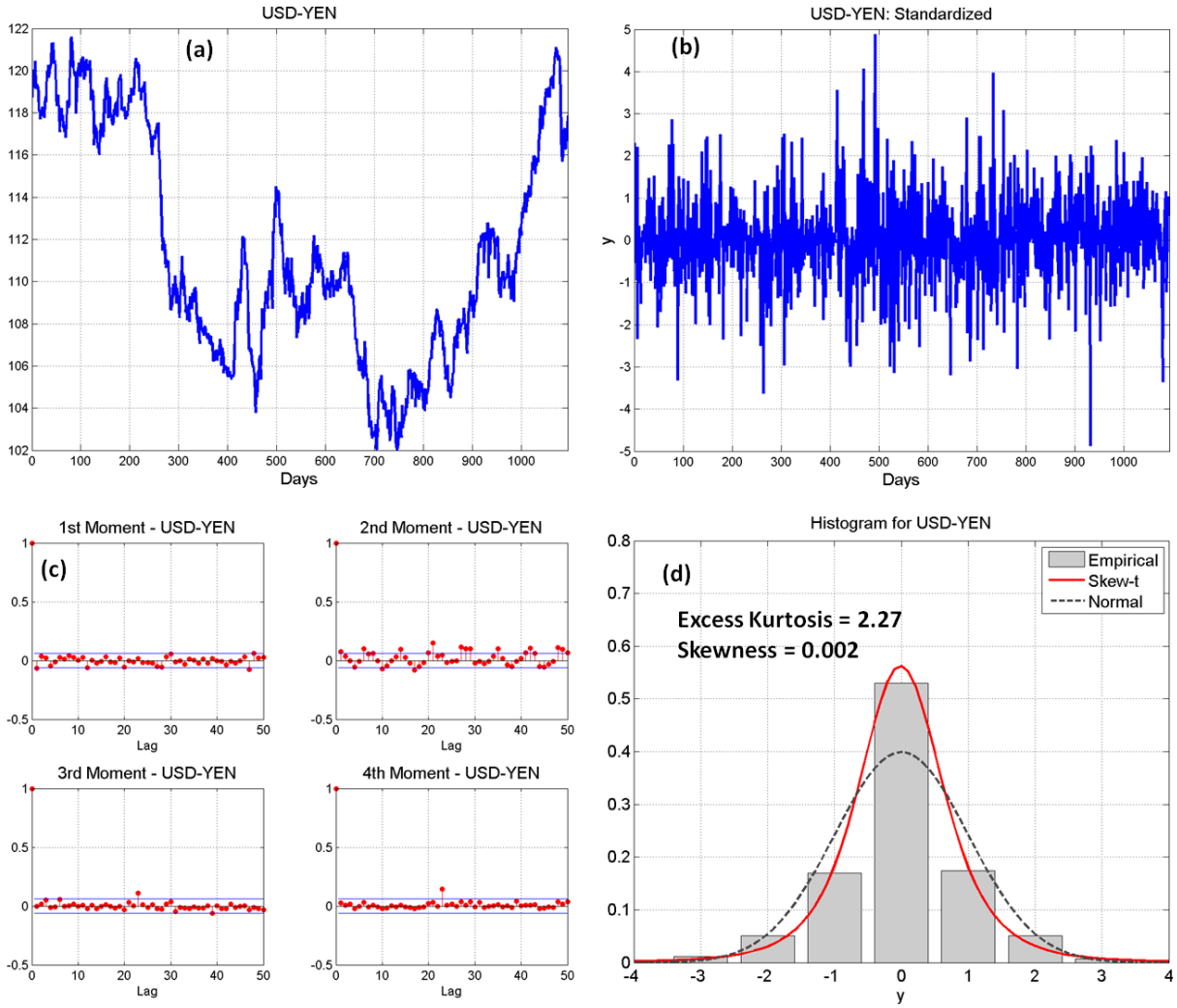

Figure 4.4: US Dollar denominated Japanese Yen Exchange Spot Rates: Data Examination: (a) The daily closing levels of exchange rate between January 2, 2003 and December 31, 2005; a total of 1095 data points. (b) The daily logarithmic returns of the exchange rates. (c) The ACFs of the first four moments of the log-returns of the exchange rates. (d) The histogram and best-fits with Gaussian and Skew-t distributions of the log-returns of the exchange rates.

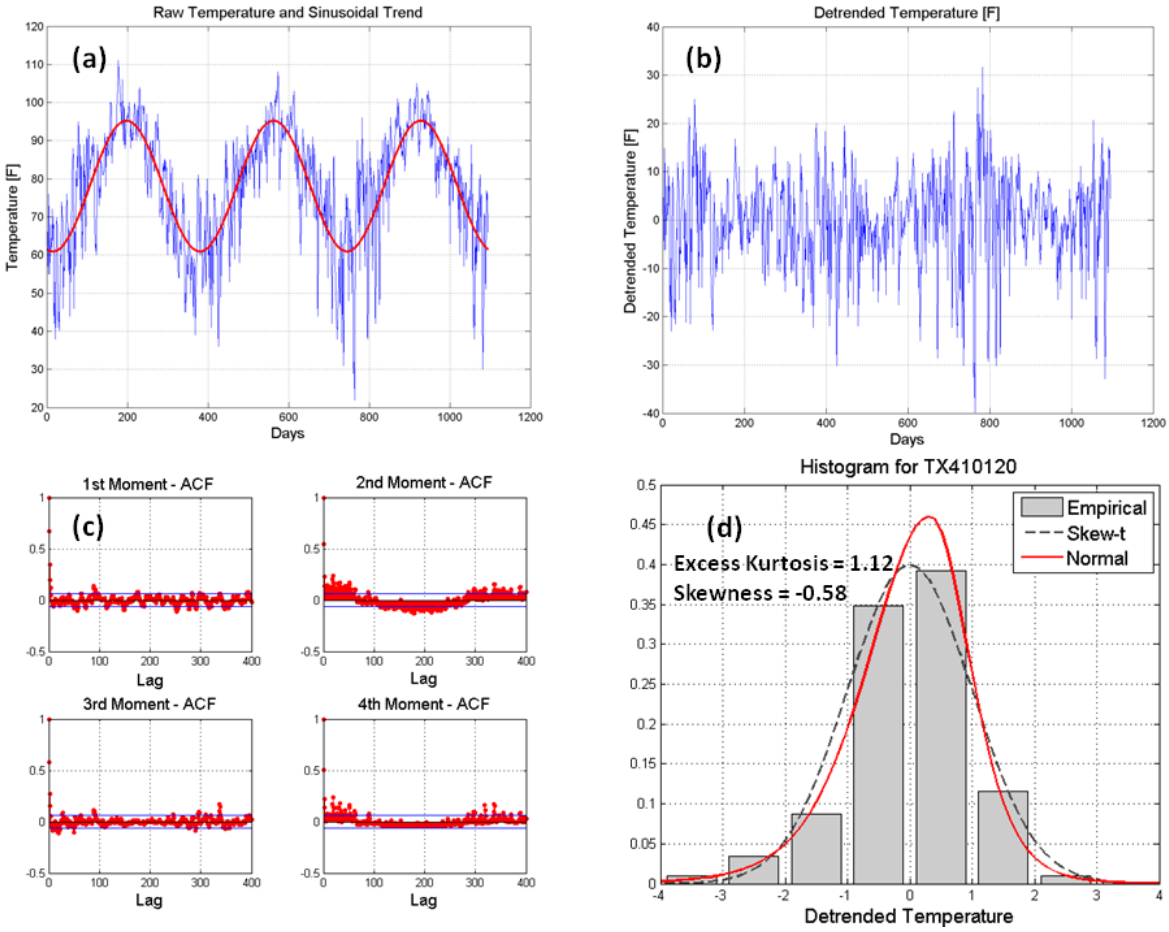

Figure 4.5: Weather station MD181750 of Maryland. Panel (a) depicts the raw maximum temperature data and the sinusoidal trend; Panel (b) shows the detrended data; Panel (c) shows the auto-correlation function of the first four moments of the detrended data; Panel (d) shows the empirical density and the estimated skew-normal distribution. The second moment ACF in (c) and the detrended temperature in (b) clearly show the periodic volatility of the residuals.

4.2.6 Results and Discussions: Univariate Time-Series Analysis

The six competing models described above are applied to each data set in turn. Learning was performed according to the plan set forth in Section 4.2.2. Figures 4.6 to 4.10 depict the results of *in-sample* learning. Recall that with negative log-likelihood, AIC, and BIC, smaller values correspond to a better goodness-of-fit. Moreover, all three metrics are closely related with different degrees of penalty for model complexity (i.e. number of parameters): the negative log-likelihood is the unpenalized version, while AIC and BIC mildly and more severely penalize the number of parameters respectively.

In general, a similar trend in relative performance is observed across all data sets. Namely, the GCSM-KCP model consistently outperforms other models on a penalized basis according to BIC. The TCSM-KCP model typically comes in as a close-second. The extra degree-of-freedom parameter associated with the Student's t-copula seems to put it at a disadvantage. This also implies that having the correct specification of the marginal distribution is much more important than having heavier tails in the copula function. In particular the GPs and TCGM-KCPs, which both have Gaussian marginals, performed relatively poorly across all data sets.

The GARCH and warped GPs also performed admirably. The warped GPs are a marked improvement from the GPs, due to the warping function taking it away from normality. However, as explained in Section 2.2.2, the allowable predictive distributions are still limited regardless of the nonlinear transformation. Further, the added complexity of the warping function is consistently penalized under the BIC metric, contributing to the relatively poor performance. On the other hand, the GARCH model performs very well with the VIX data series as expected. The simplistic GARCH(1,1) avoids severe penalties under the BIC metric. Nevertheless, it performs relatively poorly for the FX data set. Perhaps it is because to capture the seven-day periodicity in the second-moment ACF requires a GARCH with order seven or above. However, in such case, the model complexity goes up linearly and offsets any potential gain in performance under BIC.

Minor differences in ranking have also been observed in specific data sets. For instance, the TCStM-KCPs work notably better than other models for the FX data sets, which may suggest a stronger nonlinear dependency over time for the currency data. On the other hand, the GARCH(1,1) model performs best on the VIX data set, with GCStM KCP comes in a close-second. This is perhaps not a surprising result as the GARCH(1,1) model was designed specifically to capture many stylized facts of heteroskedastic series. The feedback terms in the second moment governing equation captures the volatility clustering phenomenon naturally while the KCP does not explicitly model such feature. A closer examination of the results shows interesting cases for selecting submodels within the KCP family. For the VIX and WTI data sets, note that there is a significant performance gain by choosing the skew-t and Gaussian marginal distributions, regardless of the choice of the copula function. This implies that with a simple change of marginal distribution, much of the idiosyncracies of the data can be captured. The fact that the type of copula function made little difference in the performance means that there is little complex dependency through time. On the other hand, an almost opposite conclusion can be drawn from the FX data set. In this case, choosing the skew-t marginal distribution over the Gaussian one provides only partial performance gain. Synergistic performance gain is observed if the Student's t-copula and skew-t marginal are deployed together. This is indicative of some complex long-range dependency in the FX data.

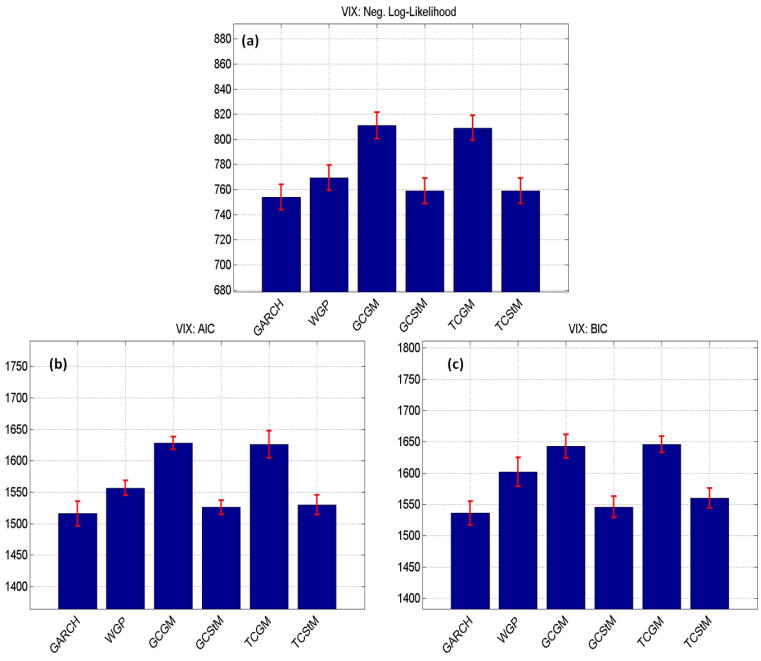

Figure 4.6: Training Results for the VIX data series: (a) Negative Log-likelihood; (b) AIC; (c) BIC. A smaller value in all above cases represents a better goodness-of-fit.

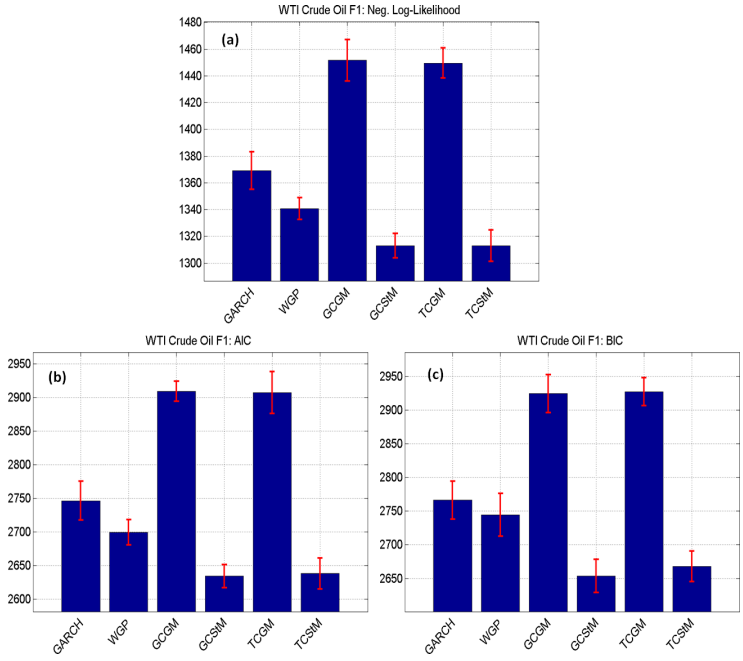

Figure 4.7: Training Results for the WTI data series: (a) Negative Log-likelihood; (b) AIC; (c) BIC. A smaller value in all above cases represents a better goodness-of-fit.

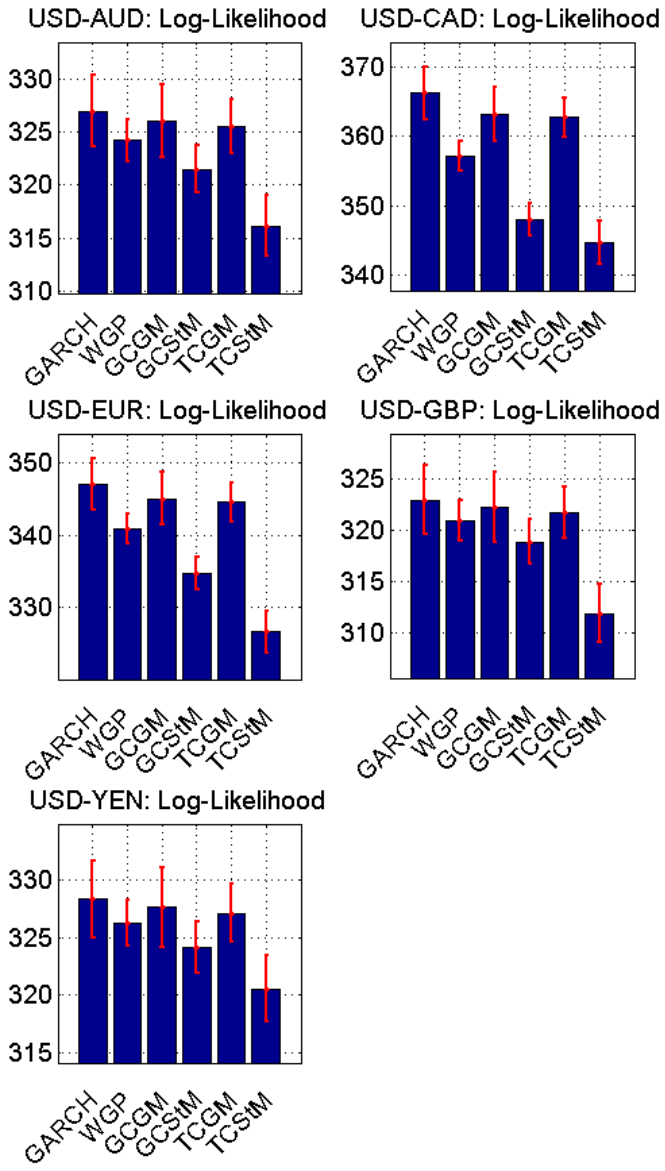

Figure 4.8: Foreign Exchange Spot Rates: Negative log-likelihood. A smaller value represents a better goodness-of-fit.

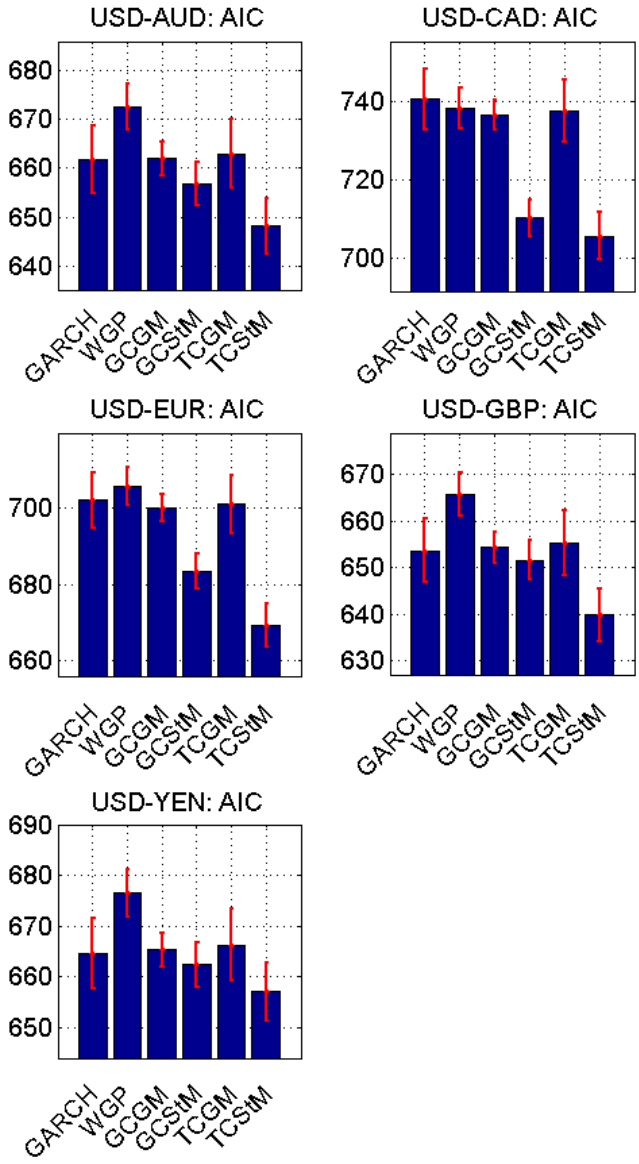

Figure 4.9: Foreign Exchange Spot Rates: AIC. A smaller value represents a better goodness-of-fit.

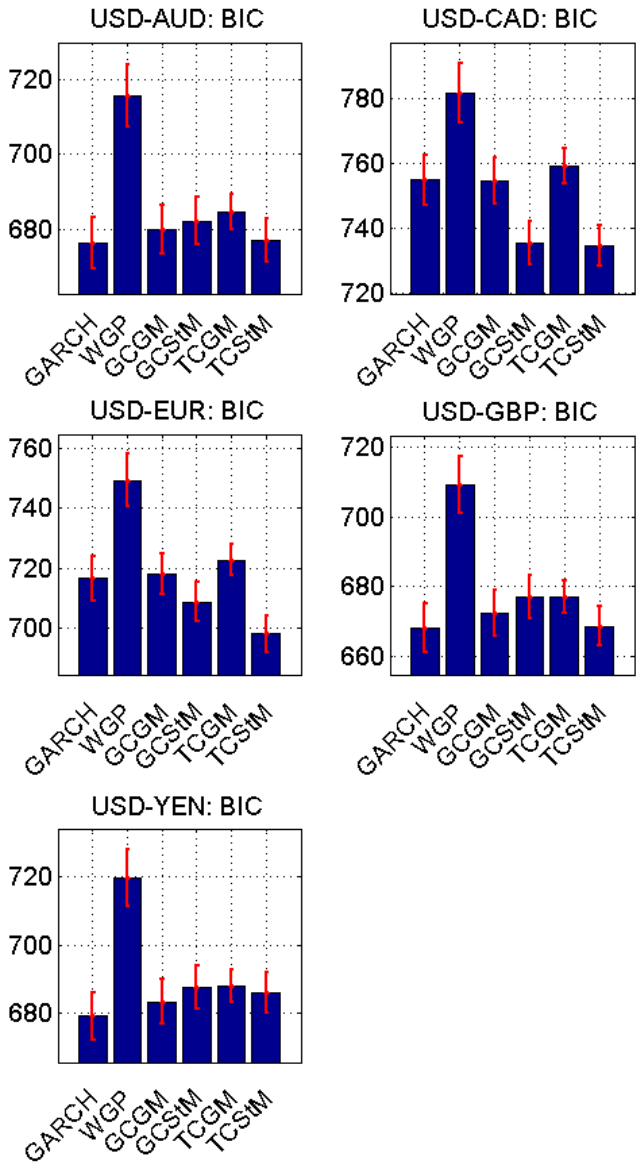

Figure 4.10: Foreign Exchange Spot Rates: BIC. A smaller value represents a better goodness-of-fit.

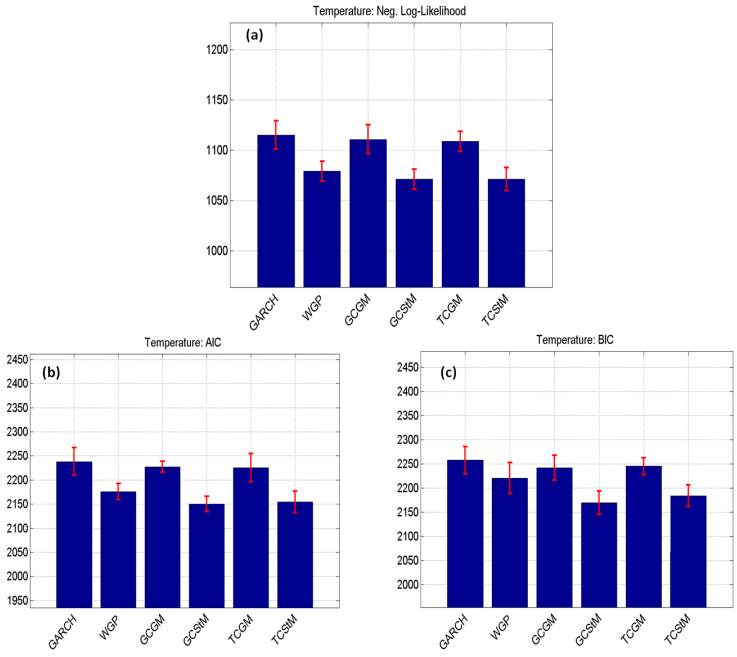

Figure 4.11: Training results for the temperature (Station MD181750 of Maryland) data series: (a) Negative Log-likelihood; (b) AIC; (c) BIC. A smaller value in all above cases represents a better goodness-of-fit.

To assess the validity of the model ranking results provided here, the likelihood ratio test (LRT) is used on a pairwise basis. The results are listed in Tables 4.2 and 4.3. The pairwise LRT was repeated for each model pair and each training window of the data sets, and the median values are listed in the tables. Recall that the test statistic for the LRT is defined as the logarithmic ratio between the maximized likelihood of the two competing models [14]:

$$\lambda_{p,q}(\mathcal{D}) = 2 \ln \frac{\mathcal{L}(\widehat{\theta_p}|\mathcal{D})}{\mathcal{L}(\widehat{\theta_q}|\mathcal{D})} \tag{4.1}$$

where the $\mathcal{L}(\widehat{\theta_p}|\mathcal{D})$ and $\mathcal{L}(\widehat{\theta_q}|\mathcal{D})$ are the likelihood functions of the models $\mathcal{M}_p$ and $\mathcal{M}_q$ evaluated on data set $\mathcal{D}$. When the likelihood functions are evaluated at the parameter sets $\widehat{\theta_p}$ and $\widehat{\theta_q}$, the likelihood functions are maximized.

The null hypothesis $\mathcal{H}_0$ for the LRT states that model $\mathcal{M}_p$ fits the data better than model $\mathcal{M}_q$. The null hypothesis is rejected when the likelihood ratio $\lambda_{p,q}$ is less than a critical value that depends on the difference in the number of parameters in the competing models. In such case, the alternate hypothesis $\mathcal{H}_a$ is accepted, which states that model $\mathcal{M}_q$ has a better goodness-of-fit than model $\mathcal{M}_p$. Please see Appendix B for a table of the critical values.

Typically, the LRT applies only to *nested models*, that is, those models in which the more complex model can be transformed into the simpler model by imposing a set of linear constraints on the parameters [14]. The set of KCPs (including the GPs) satistfy this condition, but not GARCH and WGP. Nevertheless the LRT provides further confirmation of the validity of the model ranking according to AIC/BIC. Note, the critical values for the LRT are only defined when the number of parameters in model $\mathcal{M}_p$ is larger than in model $\mathcal{M}_q$. Thus LRT was performed on only those pairwise combinations, and they are listed in Tables 4.2 and 4.3. The cases where the null hypothesis cannot be rejected are denoted by $(\mathcal{H}_0)$. For example, the TCSM-KCP performs significantly better than all other models on the currency data set as the null hypothesis cannot be rejected. The

Table 4.2: Likelihood ratio tests for the VIX, WTI and temperature data sets. Medians of pair-wise likelihood ratios among different models. The corresponding median p-values are shown inside parenthesizes. Note, the LRTs are only conducted for the pairwise combinations with model $\mathcal{M}_p$ having a larger parameter set than model $\mathcal{M}_q$. The tests that fail to reject the null hypothesis at 95% confidence level are denoted with $(\mathcal{H}_0)$. Alternately, the lack of $(\mathcal{H}_0)$ mark means the alternative hypothesis was accepted for those tests.

VIX						
	p_{GARCH}	p_{WGP}	p_{GP}	p_{GCSM}	p_{TCGM}	p_{TCSM}
q_{GARCH}	-	-15.41 (1.00)	-	-5.50 (1.00)	-55.05 (1.00)	-5.00 (1.00)
q_{WGP}	-	-	-	-	-	-
q_{GP}	57.05 (0.00) $(\mathcal{H}_0)$	41.64 (0.00) $(\mathcal{H}_0)$	-	51.55 (0.00) $(\mathcal{H}_0)$	2.15 (0.35)	52.05 (0.00) $(\mathcal{H}_0)$
q_{GCSM}	-	-9.91 (1.00)	-	-	-50.05 (1.00)	0.50 (0.48)
q_{TCGM}	-	39.64 (0.00) $(\mathcal{H}_0)$	-	-	-	50.05 (0.00) $(\mathcal{H}_0)$
q_{TCSM}	-	-10.41 (1.00)	-	-	-	-
WTI						
	p_{GARCH}	p_{WGP}	p_{GP}	p_{GCSM}	p_{TCGM}	p_{TCSM}
q_{GARCH}	-	28.46 (0.00) $(\mathcal{H}_0)$	-	56.19 (0.00) $(\mathcal{H}_0)$	-80.45 (1.00)	56.29 (0.00) $(\mathcal{H}_0)$
q_{WGP}	-	-	-	-	-	-
q_{GP}	82.45 (0.00) $(\mathcal{H}_0)$	110.92 (0.00) $(\mathcal{H}_0)$	-	138.65 (0.00) $(\mathcal{H}_0)$	2.01 (0.36)	38.75 (0.00) $(\mathcal{H}_0)$
q_{GCSM}	-	-27.73 (1.00)	-	-	-136.65 (1.00)	0.10 (0.75)
q_{TCGM}	-	108.92 (0.00) $(\mathcal{H}_0)$	-	-	-	136.75 (0.00) $(\mathcal{H}_0)$
q_{TCSM}	-	-27.83 (1.00)	-	-	-	-
Temperature						
	p_{GARCH}	p_{WGP}	p_{GP}	p_{GCSM}	p_{TCGM}	p_{TCSM}
q_{GARCH}	-	4.32 (0.83)	-4.61 (1.00)	4.63 (0.20)	-5.16 (1.00)	5.06 (0.28)
q_{WGP}	-	-	-	-	-	-
q_{GP}	-	3.98 (0.68)	-	5.41 (0.02) $(\mathcal{H}_0)$	0.02 (0.99)	6.05 (0.05) $(\mathcal{H}_0)$
q_{GCSM}	-	-3.69 (1.00)	-	-	-5.63 (1.00)	0.01 (0.92)
q_{TCGM}	-	-1.31 (1.00)	-	-	-	5.86 (0.01) $(\mathcal{H}_0)$
q_{TCSM}	-	-3.45 (1.00)	-	-	-	-

reverse is also true. For instance, the GARCH model performs better than all other models on the VIX data set as the null hypothesis is rejected when compared with all other models.

Overall, the LRT confirms the statistical significance of the model ranking results provided by AIC and BIC.

Table 4.3: Likelihood ratio tests for the currency data sets. Medians of pair-wise likelihood ratios among different models. The corresponding median p-values are shown inside parenthesizes. Note, the LRTs are only conducted for the pairwise combinations with model $\mathcal{M}_p$ having a larger parameter set than model $\mathcal{M}_q$. The tests that fail to reject the null hypothesis at 95% confidence level are denoted with $(\mathcal{H}_0)$. Alternately, the lack of $(\mathcal{H}_0)$ mark means the alternative hypothesis was accepted for those tests.

USD-AUD						
	p_{GARCH}	p_{WGP}	p_{GP}	p_{GCSM}	p_{TCGM}	p_{TCSM}
q_{GARCH}	-	2.72 (0.95)	0.90 (0.64)	5.44 (0.25)	1.40 (0.84)	10.78 (0.06)
q_{WGP}	-	-	-	-	-	-
q_{GP}	-	1.81 (0.94)	-	4.53 (0.10)	0.50 (0.78)	9.87 (0.02) $(\mathcal{H}_0)$
q_{GCSM}	-	-2.72 (1.00)	-	-	-4.03 (1.00)	5.34 (0.02) $(\mathcal{H}_0)$
q_{TCGM}	-	1.31 (0.86)	-	-	-	9.37 (0.01) $(\mathcal{H}_0)$
q_{TCSM}	-	-8.06 (1.00)	-	-	-	-
USD-CAD						
	p_{GARCH}	p_{WGP}	p_{GP}	p_{GCSM}	p_{TCGM}	p_{TCSM}
q_{GARCH}	-	9.09 (0.33)	3.03 (0.22)	18.19 (0.01) $(\mathcal{H}_0)$	3.53 (0.47)	21.49 (0.00) $(\mathcal{H}_0)$
q_{WGP}	-	-	-	-	-	-
q_{GP}	-	6.06 (0.42)	-	15.16 (0.01) $(\mathcal{H}_0)$	0.50 (0.78)	18.46 (0.01) $(\mathcal{H}_0)$
q_{GCSM}	-	-9.12 (1.00)	-	-	-14.66 (1.00)	3.30 (0.07)
q_{TCGM}	-	5.56 (0.23)	-	-	-	17.96 (0.00) $(\mathcal{H}_0)$
q_{TCSM}	-	-12.39 (1.00)	-	-	-	-
USD-EUR						
	p_{GARCH}	p_{WGP}	p_{GP}	p_{GCSM}	p_{TCGM}	p_{TCSM}
q_{GARCH}	-	6.18 (0.63)	2.06 (0.36)	12.37 (0.02) $(\mathcal{H}_0)$	2.56 (0.63)	20.47 (0.01) $(\mathcal{H}_0)$
q_{WGP}	-	-	-	-	-	-
q_{GP}	-	4.12 (0.66)	-	10.30 (0.01) $(\mathcal{H}_0)$	0.53 (0.76)	18.41 (0.01) $(\mathcal{H}_0)$
q_{GCSM}	-	-6.21 (1.00)	-	-	-9.80 (1.00)	8.10 (0.01) $(\mathcal{H}_0)$
q_{TCGM}	-	3.62 (0.46)	-	-	-	17.91 (0.01) $(\mathcal{H}_0)$
q_{TCSM}	-	-14.29 (1.00)	-	-	-	-
USD-GBP						
	p_{GARCH}	p_{WGP}	p_{GP}	p_{GCSM}	p_{TCGM}	p_{TCSM}
q_{GARCH}	-	2.01 (0.98)	0.67 (0.72)	4.02 (0.40)	1.17 (0.88)	10.97 (0.05) $(\mathcal{H}_0)$
q_{WGP}	-	-	-	-	-	-
q_{GP}	-	1.34 (0.97)	-	3.35 (0.19)	0.63 (0.73)	10.30 (0.02) $(\mathcal{H}_0)$
q_{GCSM}	-	-2.14 (1.00)	-	-	-2.85 (1.00)	6.95 (0.01) $(\mathcal{H}_0)$
q_{TCGM}	-	0.84 (0.93)	-	-	-	9.80 (0.01) $(\mathcal{H}_0)$
q_{TCSM}	-	-8.96 (1.00)	-	-	-	-
USD-YEN						
	p_{GARCH}	p_{WGP}	p_{GP}	p_{GCSM}	p_{TCGM}	p_{TCSM}
q_{GARCH}	-	2.05 (0.98)	0.69 (0.71)	4.11 (0.39)	1.19 (0.88)	7.78 (0.17)
q_{WGP}	-	-	-	-	-	-
q_{GP}	-	1.37 (0.97)	-	3.42 (0.18)	0.76 (0.68)	7.10 (0.07) $(\mathcal{H}_0)$
q_{GCSM}	-	-2.05 (1.00)	-	-	-2.92 (1.00)	3.67 (0.06)
q_{TCGM}	-	0.87 (0.93)	-	-	-	6.60 (0.04) $(\mathcal{H}_0)$
q_{TCSM}	-	-5.73 (1.00)	-	-	-	-

To take a closer look at the learned models, a series of plots are generated, as shown in Figure 4.12, for each data set. The left column in the figure shows the empirical histogram of the data, and the marginal distributions learned by the competing models (to maintain visual simplicity, only the learned distributions from the TCSM-KCP, WGP, and GARCH models are shown). A perfect match is not expected since the interdependency among samples in the sample path will distort the fitting in the marginal distribution in this manner. In addition, the empirical histogram is not strictly a correct representation of the marginal distributions of the true data generating process. Nevertheless, examining these plots would show us if the model has learned some important features of the underlying data. In general, enabled by the ability to model heavy-tail, kurtosis, and skewness, the KCP model seems to have learned those aspects of the data better than the warped GP and GARCH model.

On a related note, the right column in Figure 4.12 shows the warping function learned by the warp GP model. It is interesting to note that the choice of initial values of parameters must be carefully chosen when training the warped GPs. When the parameters are initialized randomly, the kernel functions tend to learn a narrow bandwidth parameter with a large noise density parameter, probably trapped at a local minimum. To learn a set of sensible parameters, we used the MLE parameters of the GPs, as the initial values of the common parameters for the warp GPs while initializing the rest of the parameters randomly. The resulting parameters are sensible, with the expected superior performance over GPs.

Next we shift gear to examine the *predictive* performance metric: PITs, which are shown in Figures 4.13 to 4.20. In these figures, the left columns show the PIT-transformed series of the 1-day-ahead predictions made by each model. The closer it is to a uniform distribution, the better the model performed. The Anderson-Darling (AD) test was also performed on each instance to properly accept or reject a model on each data series. Table 4.4 lists the Anderson-Darling scores of each data set for each model. The right

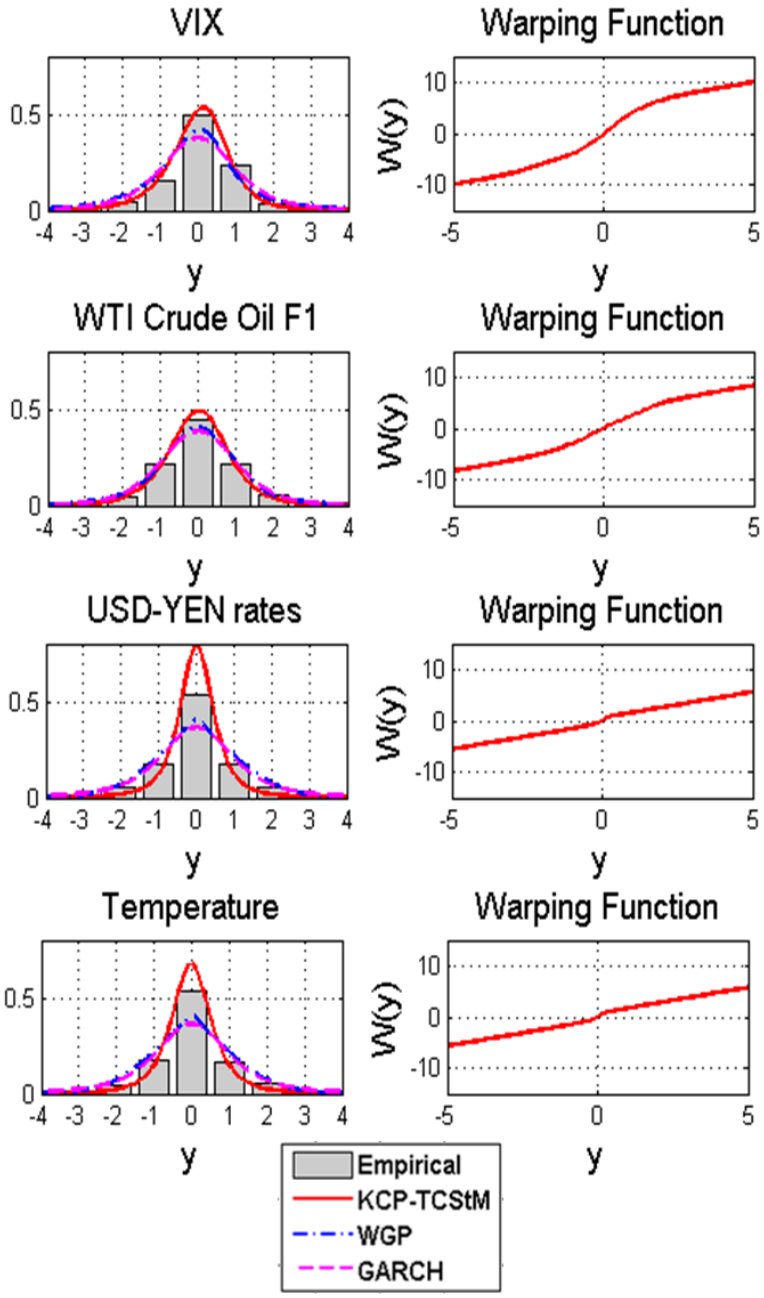

Figure 4.12: Some insights into the learned models for all data sets. The USD-YEN exchange rates and data from the weather station MD181750 of Maryland are used as representatives for the foreign exchange rates and temperature data sets. Left column: The marginal distributions of the KCP-TCSM, WGP, and GARCH models with the MLE parameters are shown with the data histogram. Note, the WGP under-estimated the positive skew of the distribution, while the GARCH model is unable to model the skew. The KCP model with the skew-t distribution seems to match the empirical distribution best. Right column: The warping function that was learned with WGP.

columns show the first-order ACF of the PIT-transformed series. It serves as a visual check of sample independence. Note that the GCSM and the TCSM KCPs tend to have a PIT histogram closest to the uniform distribution than the other models, while all ACFs shown virtual no serial correlation. This is indicative of a superior goodness-of-fit. It is also in general agreement with the model ranking given by AIC and BIC earlier. In most cases, the model mismatch as illustrated by the PIT histograms show a peak near the middle of the distribution which implies the competing models are typically under-precise (see Figure 3.10), i.e. they are unable to the excess kurtosis of the underlying DGPs.

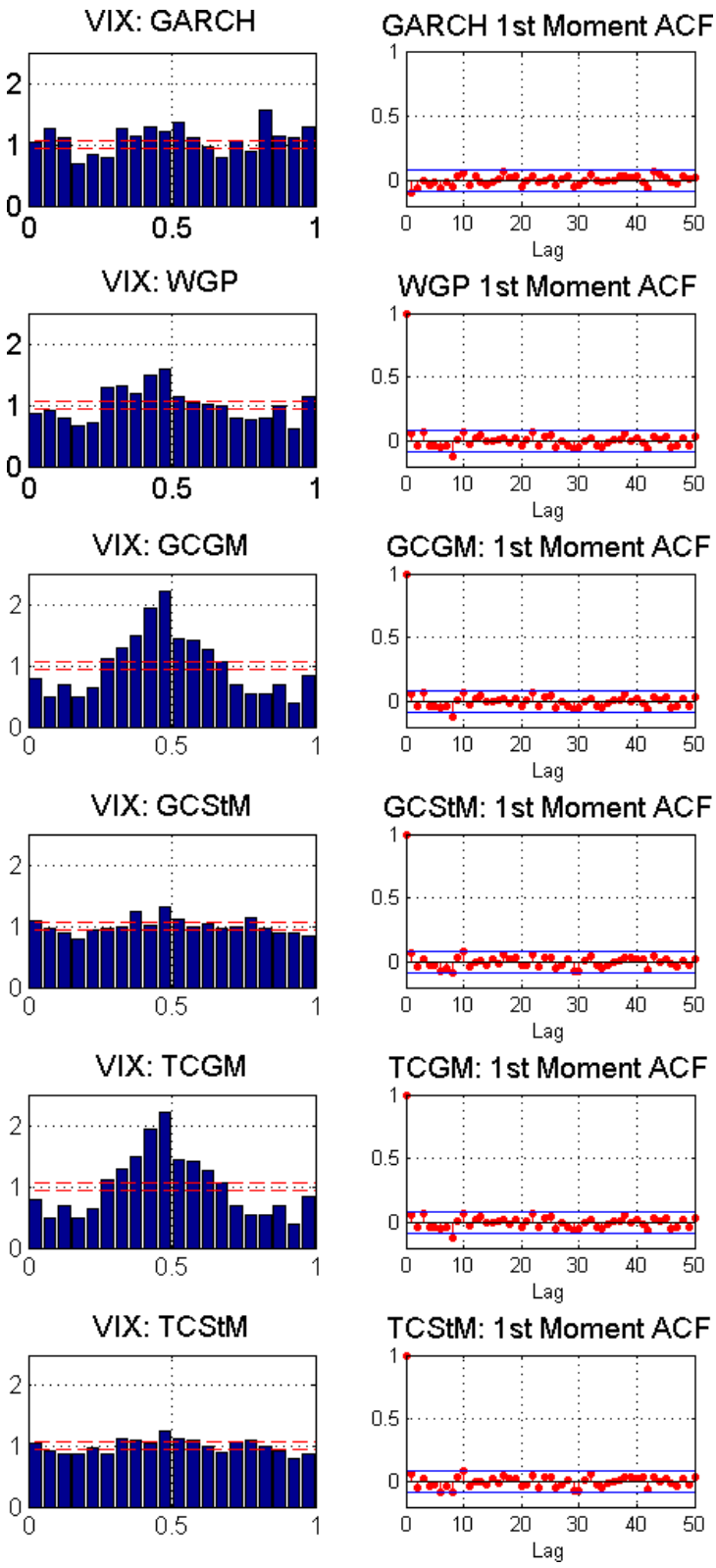

Figure 4.13: The VIX data set. Left column: PITs produced by learning the data using a 100 day sliding window to perform a one-day-ahead prediction. Right column: ACF of the corresponding PIT series as a visual independence test of the PIT series.

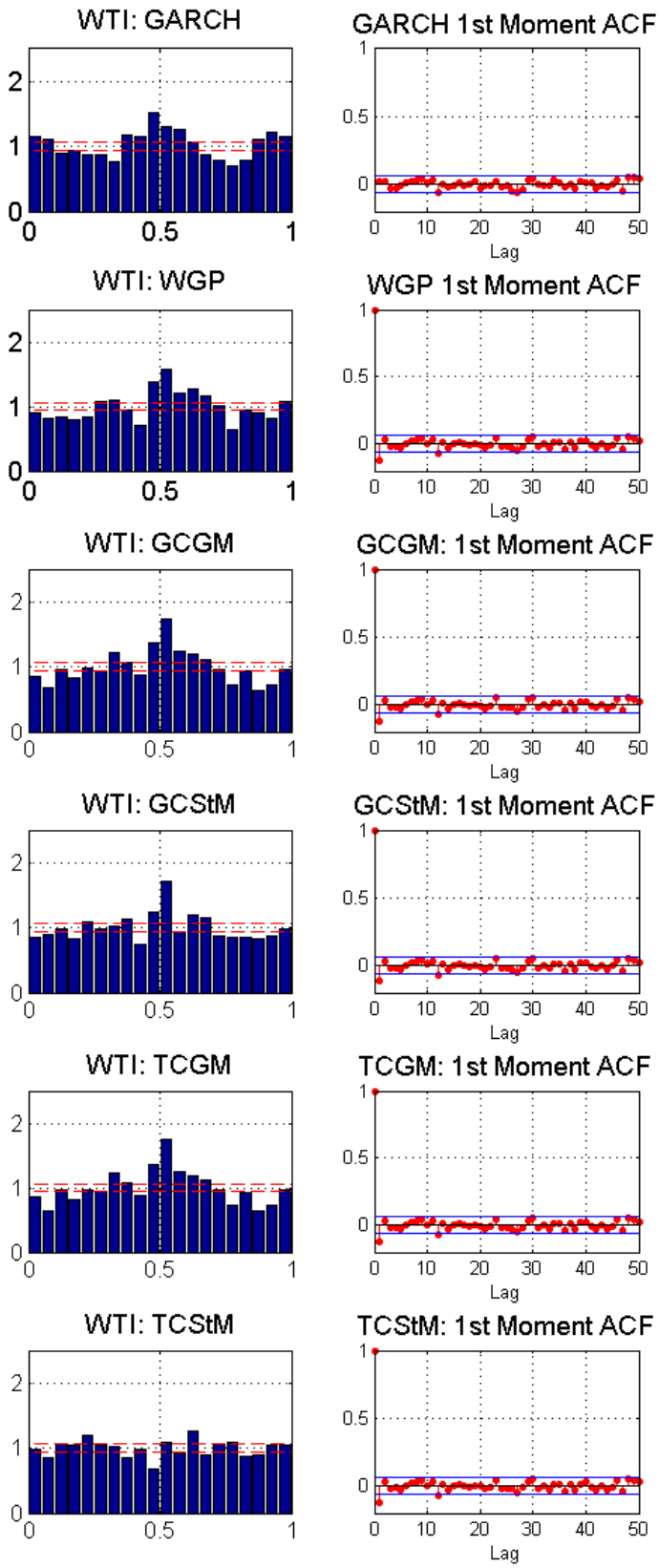

Figure 4.14: The WTI data set. Left column: PITs produced by learning the data using a 100 day sliding window to perform a one-day-ahead prediction. Right column: ACF of the corresponding PIT series as a visual independence test of the PIT series.

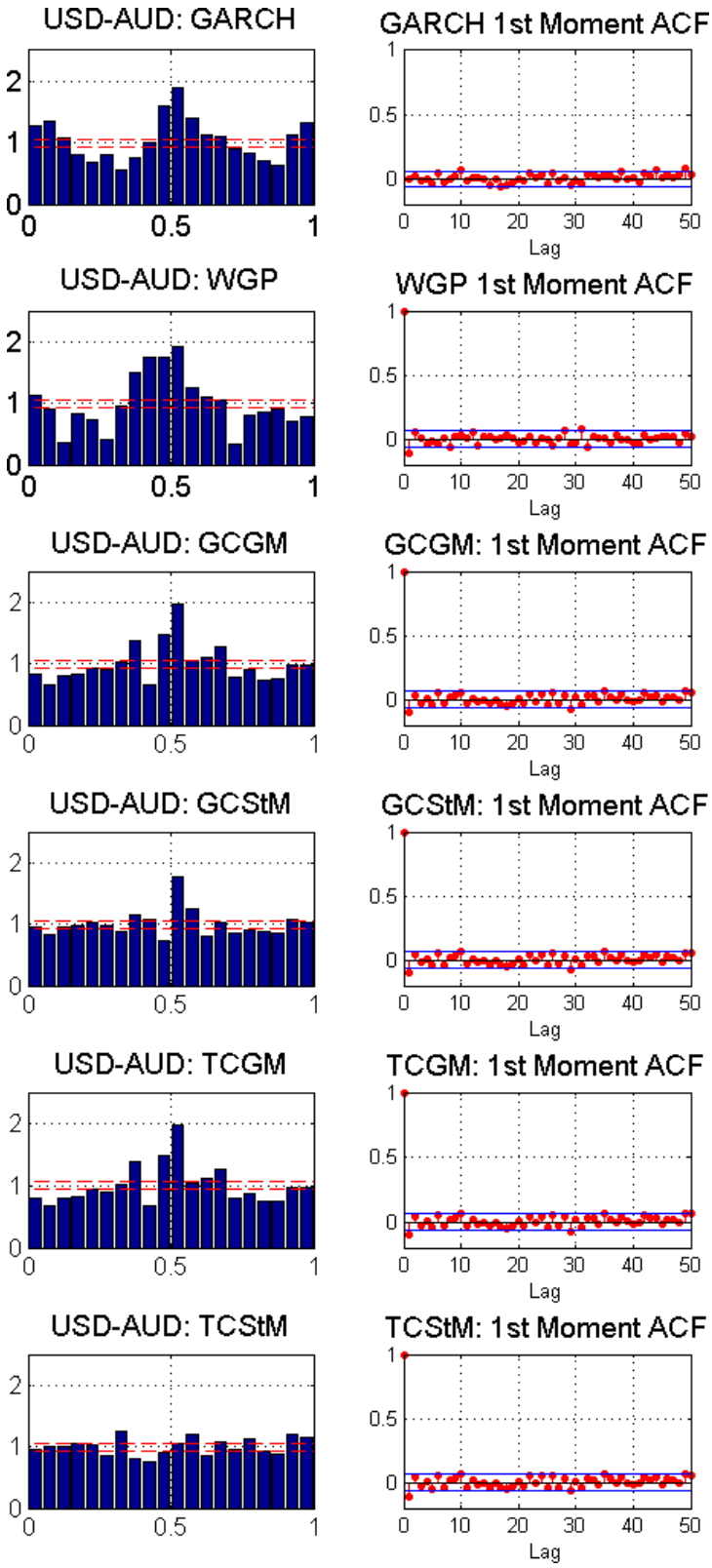

Figure 4.15: The AUD data set. Left column: PITs produced by learning the data using a 100 day sliding window to perform a one-day-ahead prediction. Right column: ACF of the corresponding PIT series as a visual independence test of the PIT series.

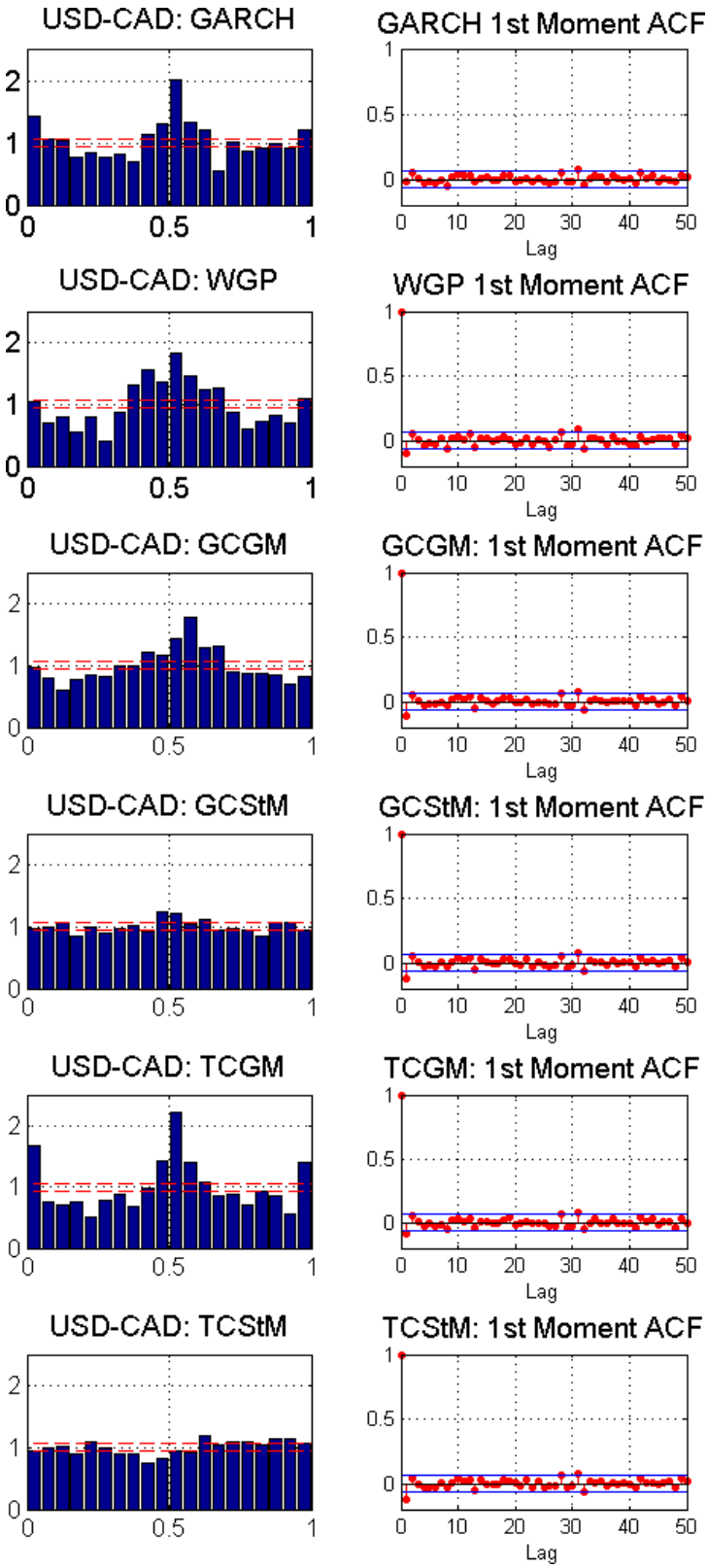

Figure 4.16: The CAD data set. Left column: PITs produced by learning the data using a 100 day sliding window to perform a one-day-ahead prediction. Right column: ACF of the corresponding PIT series as a visual independence test of the PIT series.

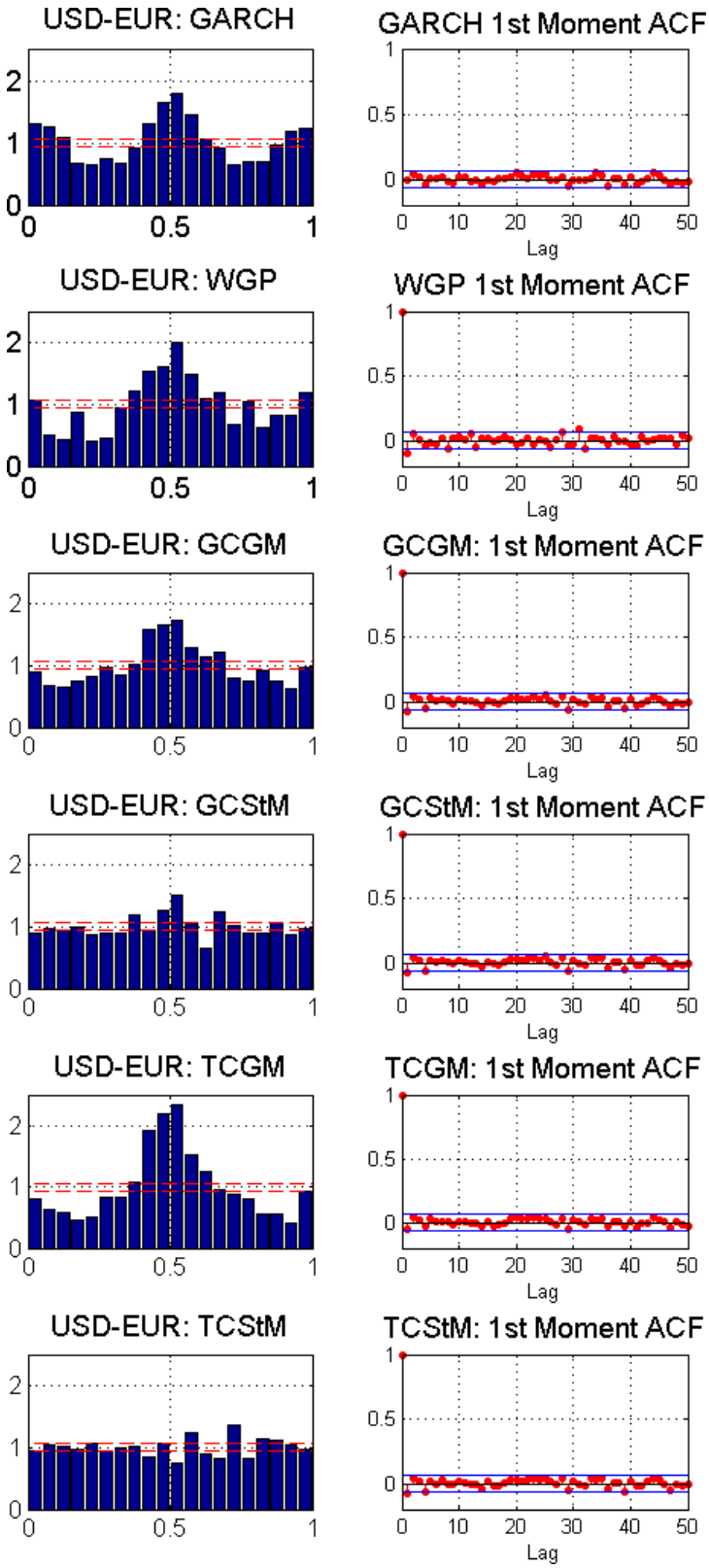

Figure 4.17: The EUR data set. Left column: PITs produced by learning the data using a 100 day sliding window to perform a one-day-ahead prediction. Right column: ACF of the corresponding PIT series as a visual independence test of the PIT series.

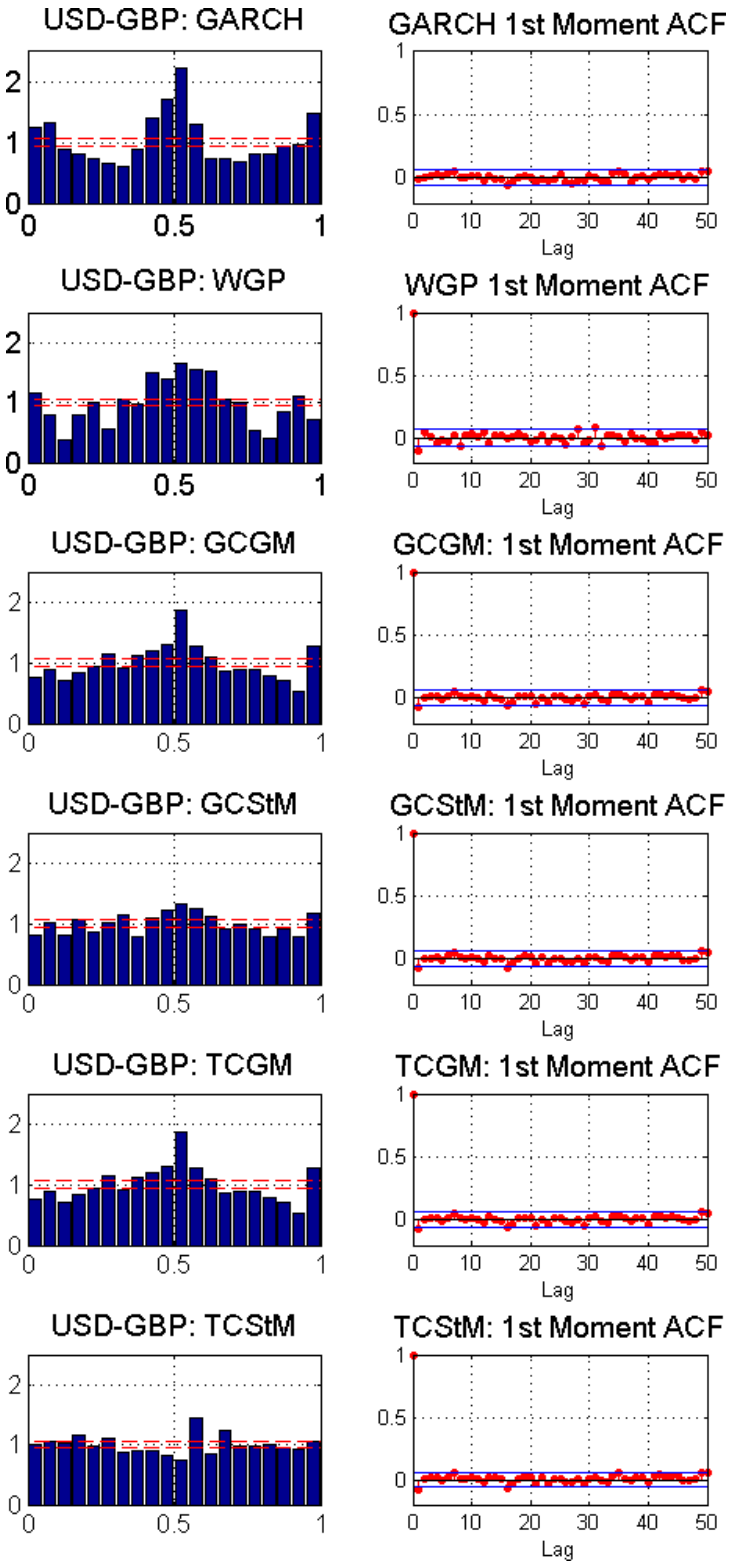

Figure 4.18: The GPB data set. Left column: PITs produced by learning the data using a 100 day sliding window to perform a one-day-ahead prediction. Right column: ACF of the corresponding PIT series as a visual independence test of the PIT series.

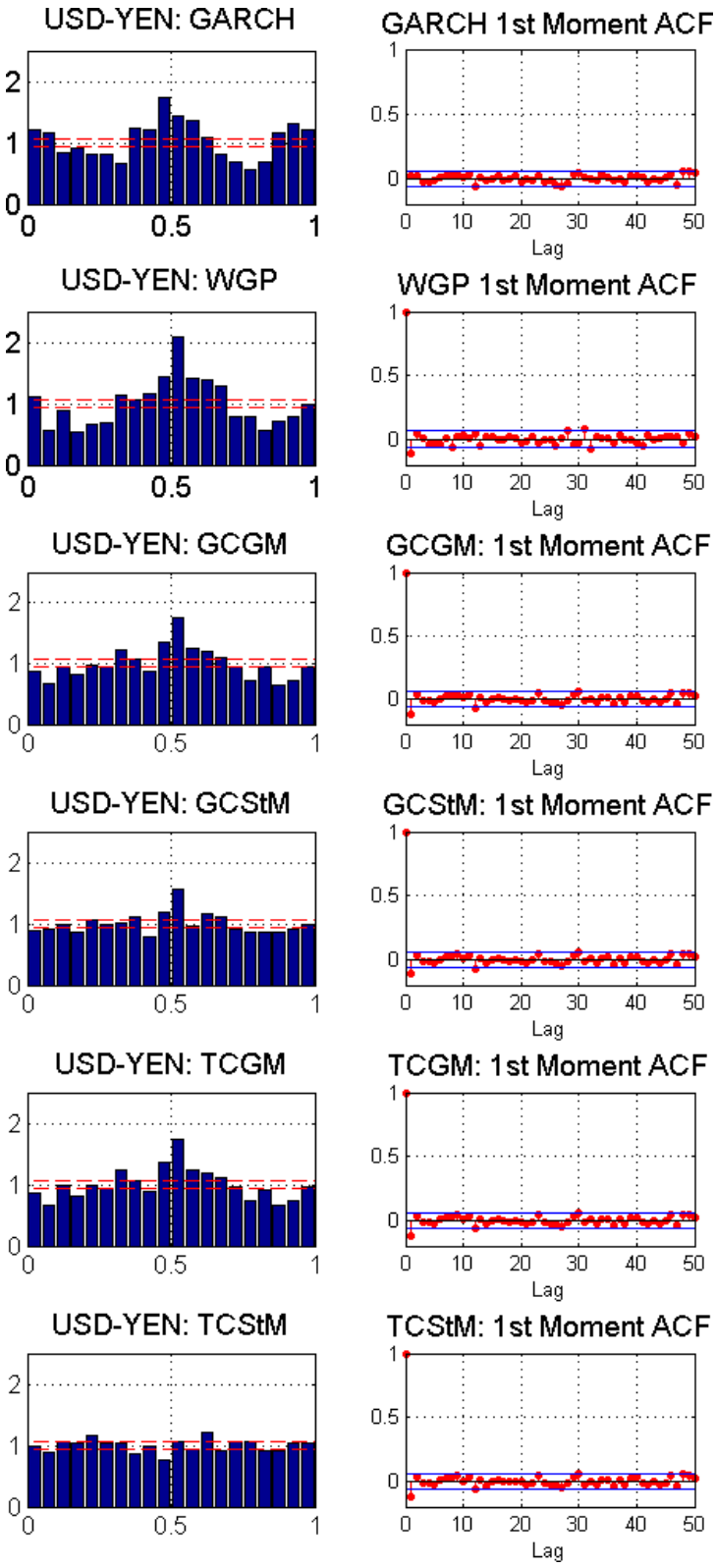

Figure 4.19: The YEN data set. Left column: PITs produced by learning the data using a 100 day sliding window to perform a one-day-ahead prediction. Right column: ACF of the corresponding PIT series as a visual independence test of the PIT series.

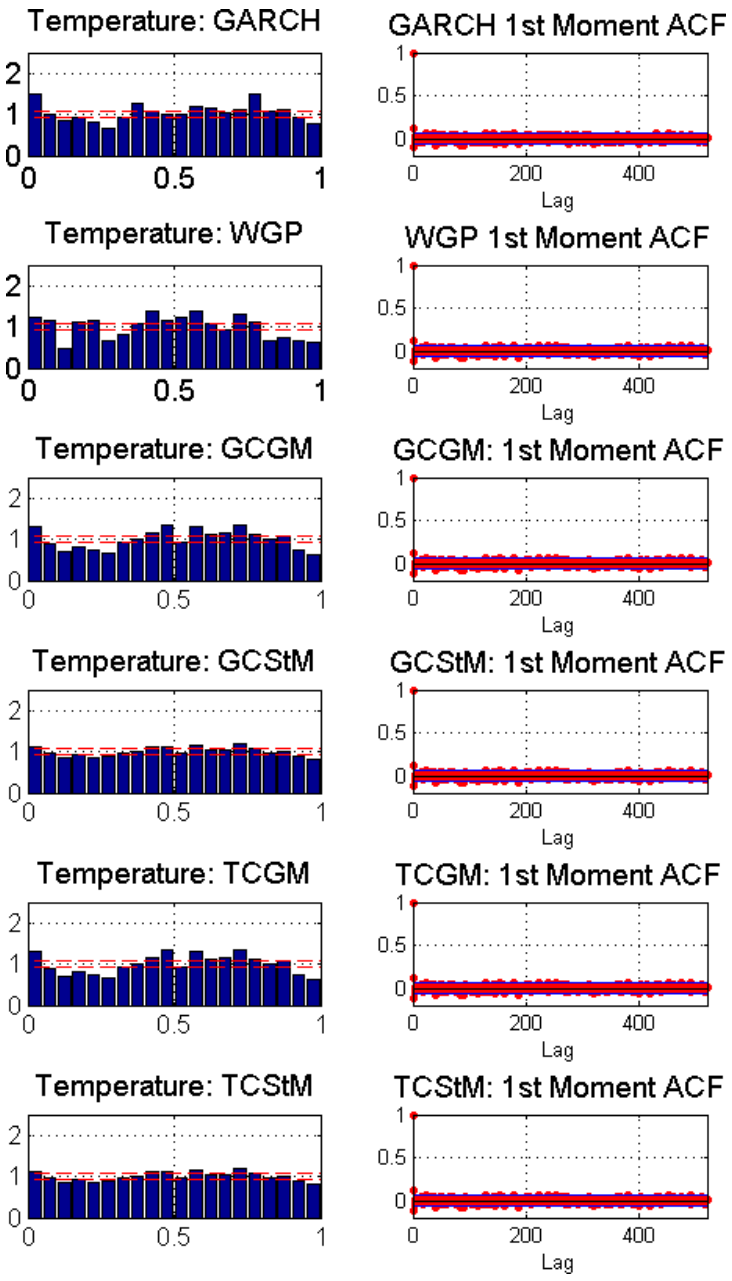

Figure 4.20: The temperature data set (Station MD181750 of Maryland). Left column: PITs produced by learning the data using a 100 day sliding window to perform a one-day-ahead prediction. Right column: ACF of the corresponding PIT series as a visual independence test of the PIT series.

Table 4.4: Anderson-Darling Scores. The tests that passed with 5% significance level (critical value of 2.492 [4]) are indicated by superscript **, the tests that passed with 10% significance level (critical value of 1.933 [4]) are indicated by supercript *, while the best score within each data set are indicated by bold typeface. The p-values of the AD test are also provided in parentheses and they are computed according to [71].

Model	VIX	WTI	AUD-USD	CAD-USD	EUR-USD	GBP-USD	YEN-USD
GARCH	0.69**	8.23	4.50	2.94	4.60	5.52	3.22
	(0.567)	(0.000)	(0.005)	(0.029)	(0.004)	(0.002)	(0.021)
WGP	6.11	6.42	10.43	12.14	9.57	12.89	10.16
	(0.001)	(0.001)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
GCGM-KCP	6.05	18.54	7.18	8.27	10.08	6.82	6.10
	(0.001)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.001)
GCSM-KCP	**0.25****	**0.29****	1.84**	**0.90****	1.81**	2.24*	2.30*
	(0.970)	(0.945)	(0.113)	(0.414)	(0.117)	(0.068)	(0.063)
TCGM-KCP	6.12	18.63	6.32	8.45	10.89	5.82	6.11
	(0.001)	(0.000)	(0.001)	(0.000)	(0.000)	(0.001)	(0.001)
TCSM-KCP	0.26**	0.31**	**1.42****	1.21**	**0.55****	**0.64****	**0.67****
	(0.965)	(0.930)	(0.197)	(0.264)	(0.696)	(0.611)	(0.584)

A closer examination of Table 4.4 shows that the GCSM-KCP model[10] is accepted by the AD tests for all data sets (with the weather data considered separately), while the TCSM-KCP is also accepted for all but the VIX data set. The GARCH model performs admirably across the board. While the warped GP model seems to have made improvements over the GPs for the VIX and WTI data sets, it seems to have performed less well for the currency data sets.

[10]GCSM in Table 4.4 collective denotes Gaussian Copula with Skew normal or t marginal depends on the data set, as detailed in the text.

For the temperature data set, we use a slightly different presentation of the univariate results, as the data set involves a large collection of weather stations across the US. First, the acceptance and rejection frequencies for all models are shown in Figure 4.21a. On the poor performing end of the spectrum, the Anderson-Darling test rejects the hypothesis that GP and TCGM KCP are sufficiently good models (with 95% confidence level) in ∼43% and ∼42% of the cases; whereas, the TCSM KCP was shown to be the best model as it was rejected in only ∼7.5% of the cases. Figures 4.21b and 4.21c show the frequency of each model being the *best* and the *worst* model for each data set as ranked by the BIC. The GCSM and TCSM KCPs clearly dominate over the GP and TCGM KCP, demonstrating that most of the modeling power (for this data set) comes from the skewness of the marginal distributions while the extra tail-dependency from the Student's t-copula provides an incremental gain. The performance of the GARCH(1,1) model with t-innovations seems to be hit-and-miss with this data set, yet it is quite impressive that it ranks first about 22% of the time, even though the KCPs use a kernel function that is custom designed for this data set. Finally, while the warped GP delivered small improvements to the GP on average, its performance is severely penalized by the model complexity (i.e. number of parameters involved).

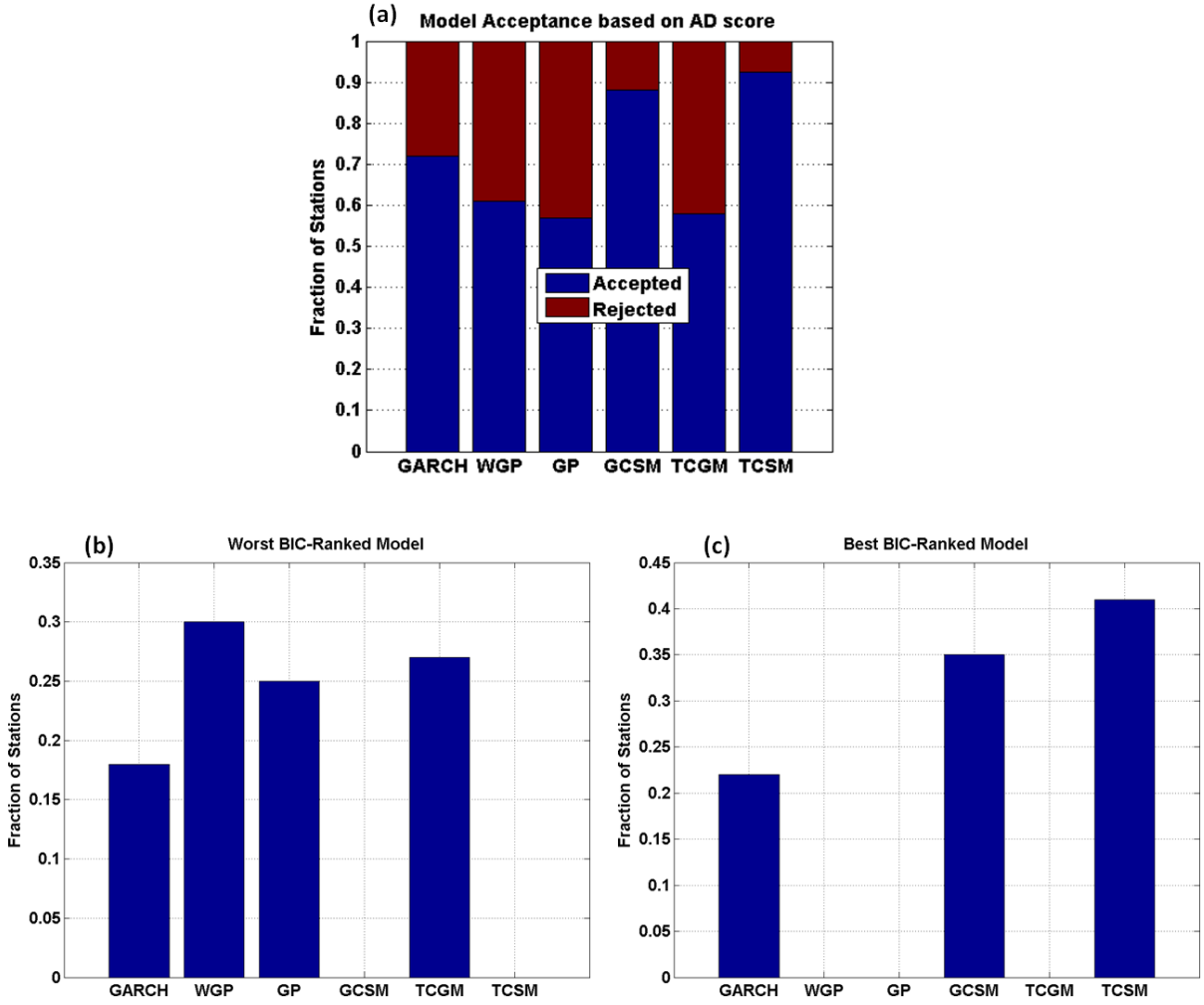

Figure 4.21: Univariate model selection results. The top panel shows the fraction of model acceptance/rejection based on the Anderson-Darling criterion; the lower panels show the fractions of each models ranked to be the most and the least appropriate model for the univariate temperature series according to the achieved BIC.

Finally, to provide additional sanity checks, Figures 4.22 to 4.25, show sample paths that are drawn from each learned model. This is not intended to be part of the metric to gauge the performance of the models as the information presented here is based on a single sample path drawn from each model. These figures are presented here to ensure that all the models, though having different modeling power, are doing something sensible. In each figure, the first column of panels from left to right show the time series, the corresponding ACF of the second moment (only for Figures 4.24 and 4.25), the histograms, and the q-q plots. The first row shows the plots of the detrended data while the subsequent rows show the samples generated from the corresponding models after learning on the same set of the detrended data. The second moment ACFs are included for the currency rates and temperature data sets in Figures 4.24 and 4.25 because there are special heteroskedastic behaviors embedded in the data series, and an appropriately fitted model should exhibit similar behavior. The histograms are presented for qualitative comparison. One noted example is the USD-YEN series in Figure 4.24, where the excess kurtosis observed in the data is clearly visible in the sample paths from the GCStM and TCStM KCPs but by none of the others. The q-q plots also provide another way to gauge the similarity between the sample paths from a model and the data series itself. A model represents a perfect fit if the quantile samples lie on the straight line. Any deviations from the straight line represent model mis-fit at different part of the distributions. For example, deviations from the ends of the straight line represent inadequacy of modeling powers at the tails of the distribution.

Recall that the same kernel functions are used for WGP, GP, and all KCPs in each data set. Specifically, the heteroskedastic kernel function (3.41) was designed particularly to model the periodic variances observed in the currency rates and temperature data series. Thus, it is not surprising that the same periodic structure in the variance is learned by all models (except the GARCH(1,1) model) as evident in the second moment ACFs in Figures 4.24 and 4.25. Even though the GARCH(1,1) model fails to learn any

appreciable structure for the currency rates data series (Figure 4.24), it was able to learn a similar structure for the temperature data set (Figure 4.25). This is a little surprising as the GARCH(1,1) is capable of only very short-term memory and the periodicity of the variance in the temperature data set is very long (1 year).

In visually examining the histograms and the q-q plots, the GCSM and TCSM KCPs again consistently perform better than the other models in all data sets. The most common mis-fitted features by the other models seem to be the tail parts of the marginal distribution. This is not surprising with GP and TCGM KCP, which are limited to Gaussian marginal by definition. However, even the GARCH model with t-innovations and the nonlinear transformation by the warped GP did not improve the fitting of the tail very much.

Finally, please note again that the series of sanity checks provided here focus on the marginal behavior of the data series and model, which does not provide a complete picture. Nevertheless, it provides more insight into the performance of each model at different facets of the real-life problems presented here. The combination of the BIC and AD scores of PIT should remain the ultimate evaluation metric as these scores provide a comprehensive evaluation of the model.

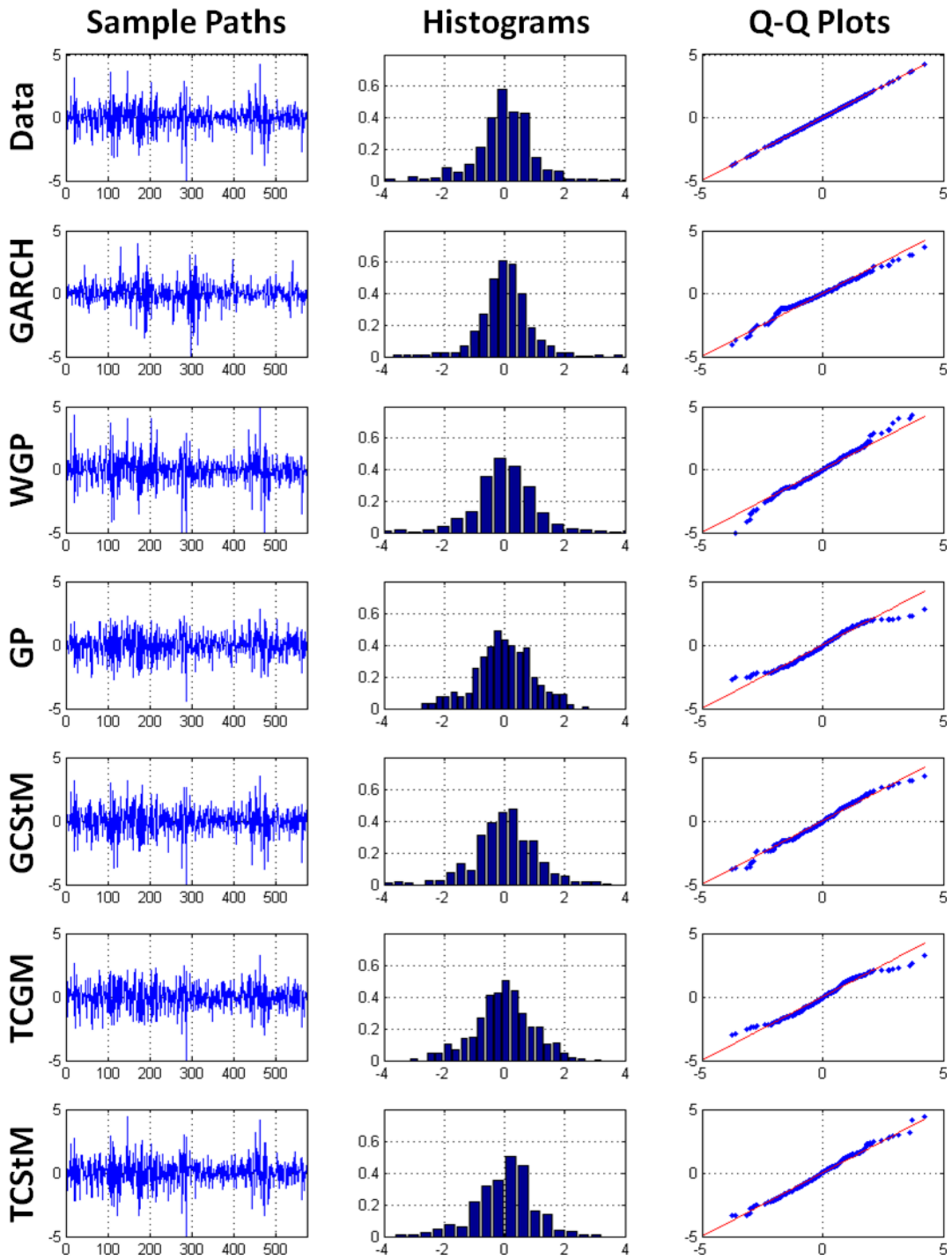

Figure 4.22: Sanity Check for the VIX data set: Left column: The actual empirical data path is shown on the top row, while sample paths drawn from the MLE models are shown below. Middle column: the corresponding histograms. Right column: the q-q plots against the empirical data.

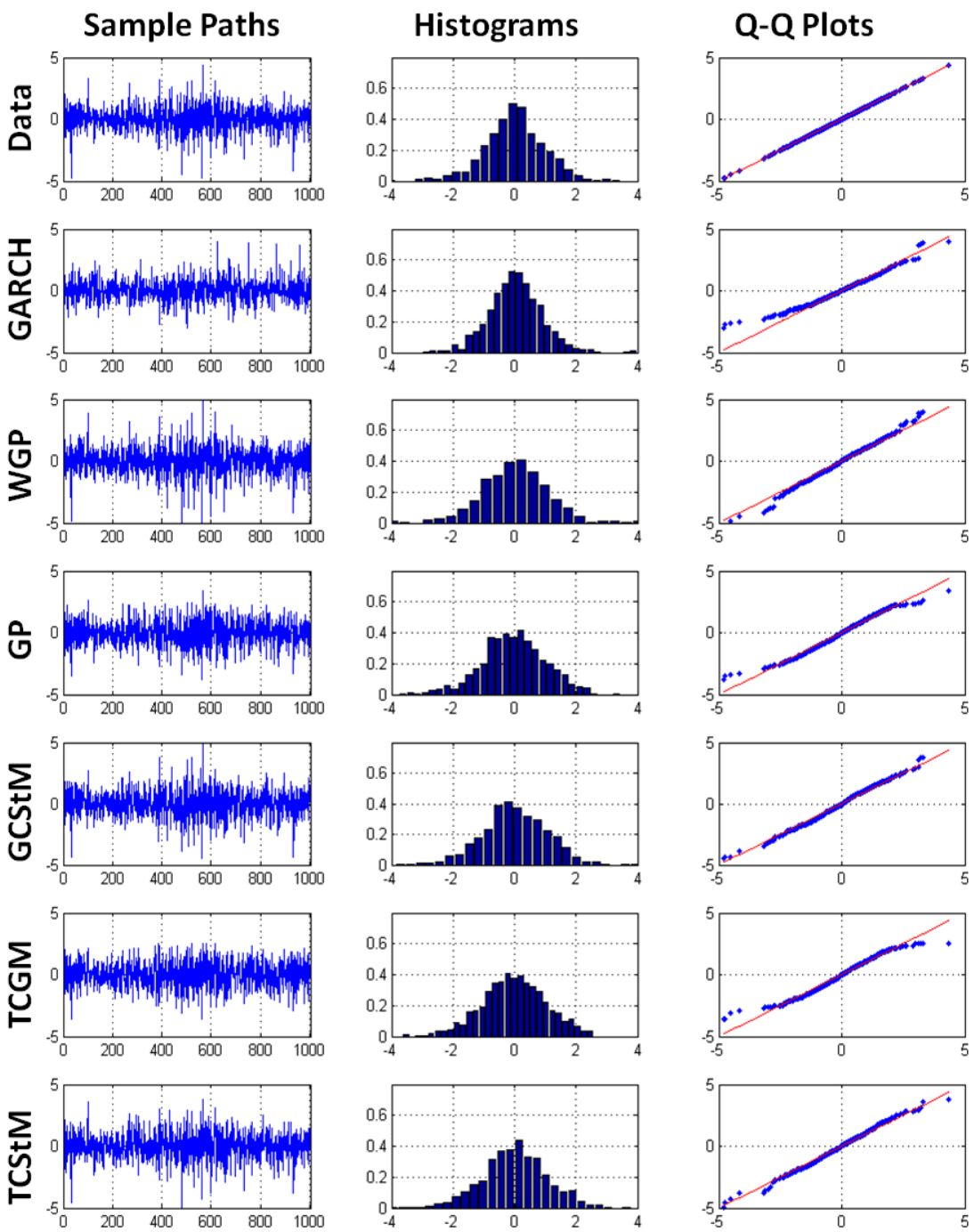

Figure 4.23: Sanity Check for the WTI data set: Left column: The actual empirical data path is shown on the top row, while sample paths drawn from the MLE models are shown below. Middle column: the corresponding histograms . Right column: the q-q plots against the empirical data.

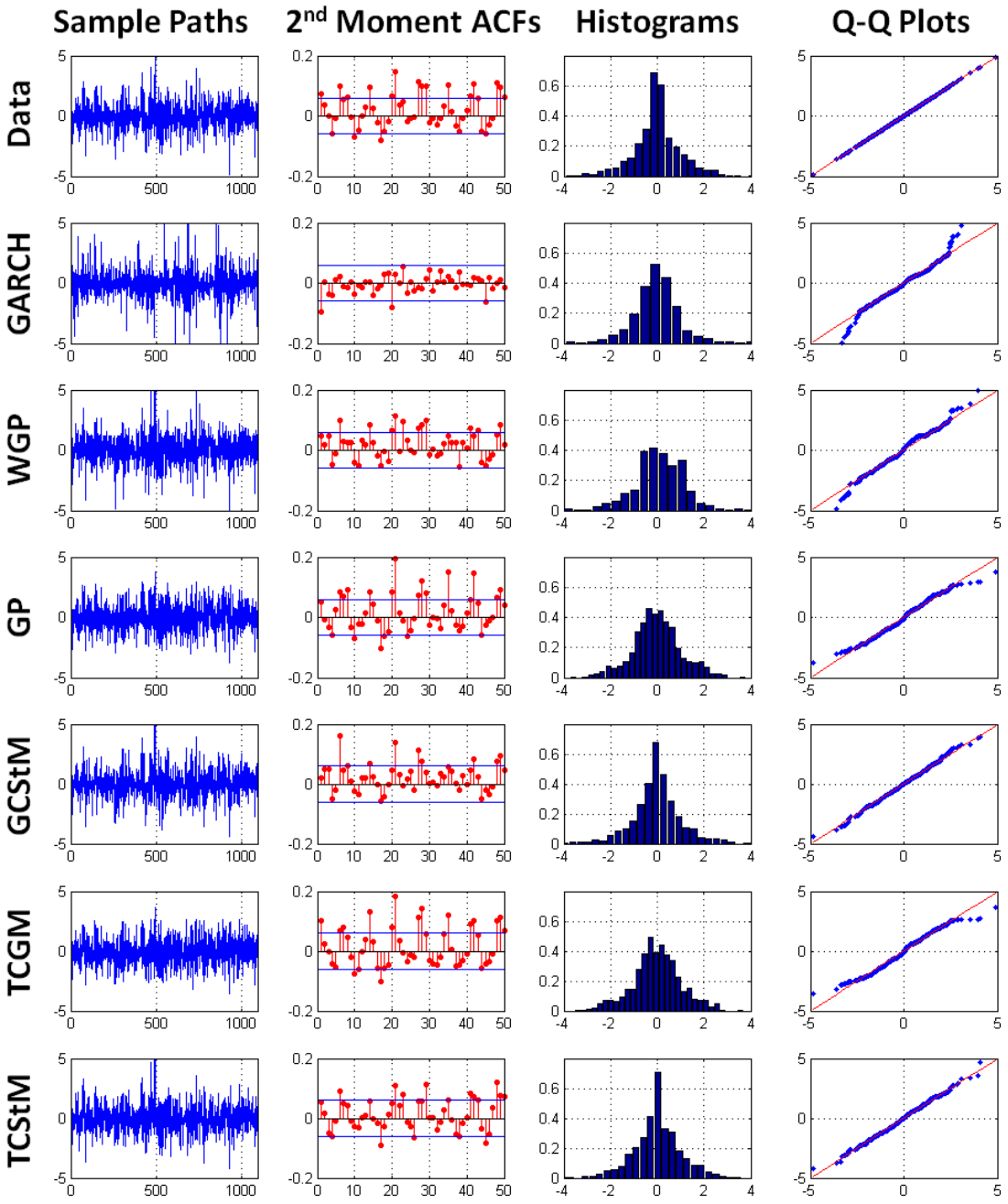

Figure 4.24: Sanity Check for the USD-Yen data set: Left column: The actual empirical data path is shown on the top row, while sample paths drawn from the MLE models are shown below. Middle column: the corresponding histograms. Right column: the q-q plots against the empirical data.

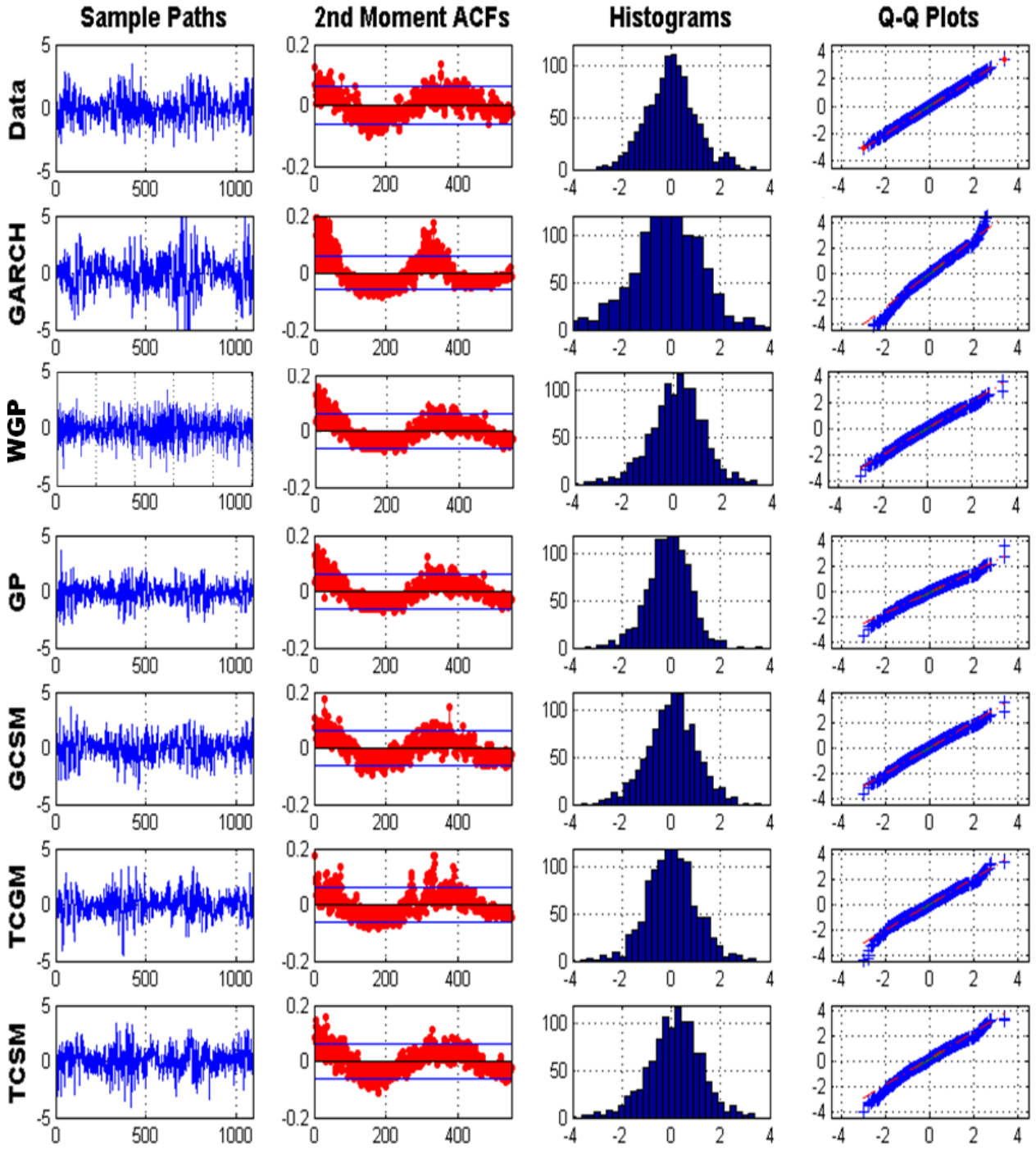

Figure 4.25: Comparing stylized facts of the detrended data and sample paths generated by models with MLE parameters based on the weather station MD181750 of Maryland.

4.2.7 Results and Discussions: Multivariate Time-Series Analysis

In this section, we study the potential codependent structure in the currency rates and temperature data sets in more detail.

First, we consider the currency rates data set. As noted earlier, with the fiat currency system, the values of each currency are no longer tied to the gold standard but freely float relative to each other according to a plethora of macroeconomic and other factors. For example, countries with similar industrial profiles (e.g. natural resources, manufacturing, service industry, etc.) may respond similarly to the same set of macroeconomical perturbations, thus resulting in correlated moves in their currency values. To this end, we consider the pairwise codependence among the five currencies. As discussed in Section 3.2, we will use another copula function, which we dubbed the *binding copula*, to model the codependent structure. In Figure 4.26, we first show the scatter-plot of the pairwise currency rates. The currency rates are first transformed with the respective empirical CDFs so the values of each data series are between 0 and 1. The resulting scatter-plot has the same support as a bivariate copula function. As shown in Figure 4.26, only four currency pairs exhibit strong dependencies and these are the AUD-CAD, AUD-EUR, EUR-CAD, and GBP-YEN pairs. No discernible dependencies are observed for the other pairs.

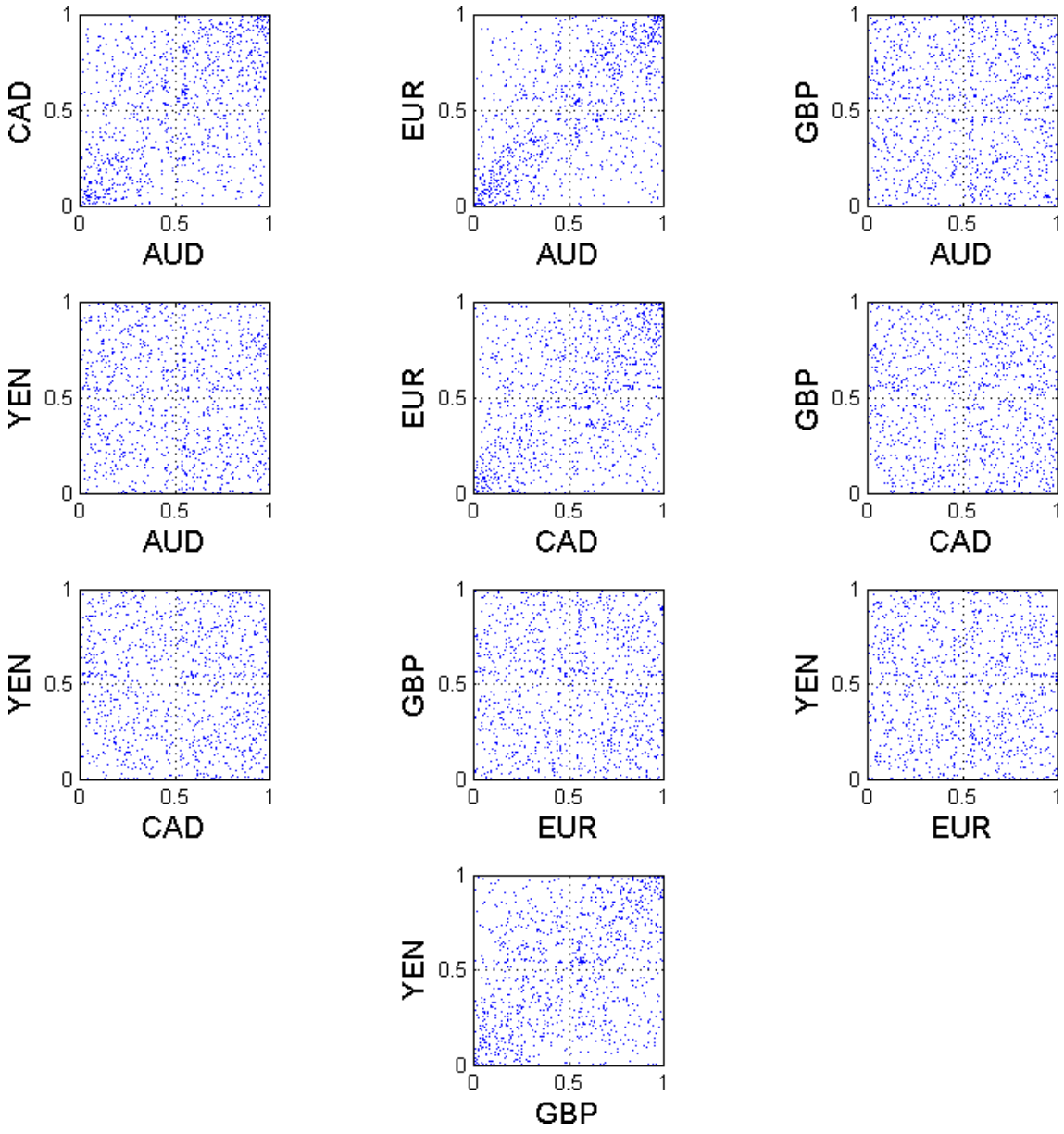

Figure 4.26: Pair-wise scatter-plots of the currency rate data set. The currency rates are first transformed with the respective empirical CDFs so the values of each data series are between 0 and 1.

Table 4.5: The ranking of bivariate binding copula for the strongly correlated currency pairs using BIC.

	FX Pairs	Copula Family [BIC]					
		Gaussian	Student's t	Frank	Gumbel	Clayton	SJC
1	AUD-CAD	-205.1	**-231.31**	-207.7	-194.41	-164.29	-205.9
2	AUD-EUR	-368.8	-370.75	**-385.33**	-359.65	-315.34	-366.3
3	CAD-EUR	-193.9	**-209.05**	-186.79	-173.05	-154.78	-172.3
4	GBP-YEN	-199.5	**-213.09**	-186.74	-192.34	-121.70	-172.6

Focusing on these four currency pairs, we proceed to choose the best binding copula for each pair. In this study, we consider the Gaussian, Student's t, Frank, Gumbel, Clayton, and Symmetrized Joe-Clayton (SJC) copulae. The best model will again be chosen by BIC. The results are tabulated in Table 4.5. It turns out that the Student's t copula is the best one to use for all but the AUD-EUR pair. The learned bivariate copula functions are shown in Figure 4.27. It is interesting to note that the Gaussian copula do not fit the currency pairs well, and as mentioned in Section 2.2.2, the GPs and the warped GPs are effectively restricted to the Gaussian copula as the binding copula when modeling multivariate time-series. Not only does this imply the superiority of the multivariate KCP model over the GPs and warped GPs for this particular data set, it also highlights the advantages that modeling flexible for different occasions.

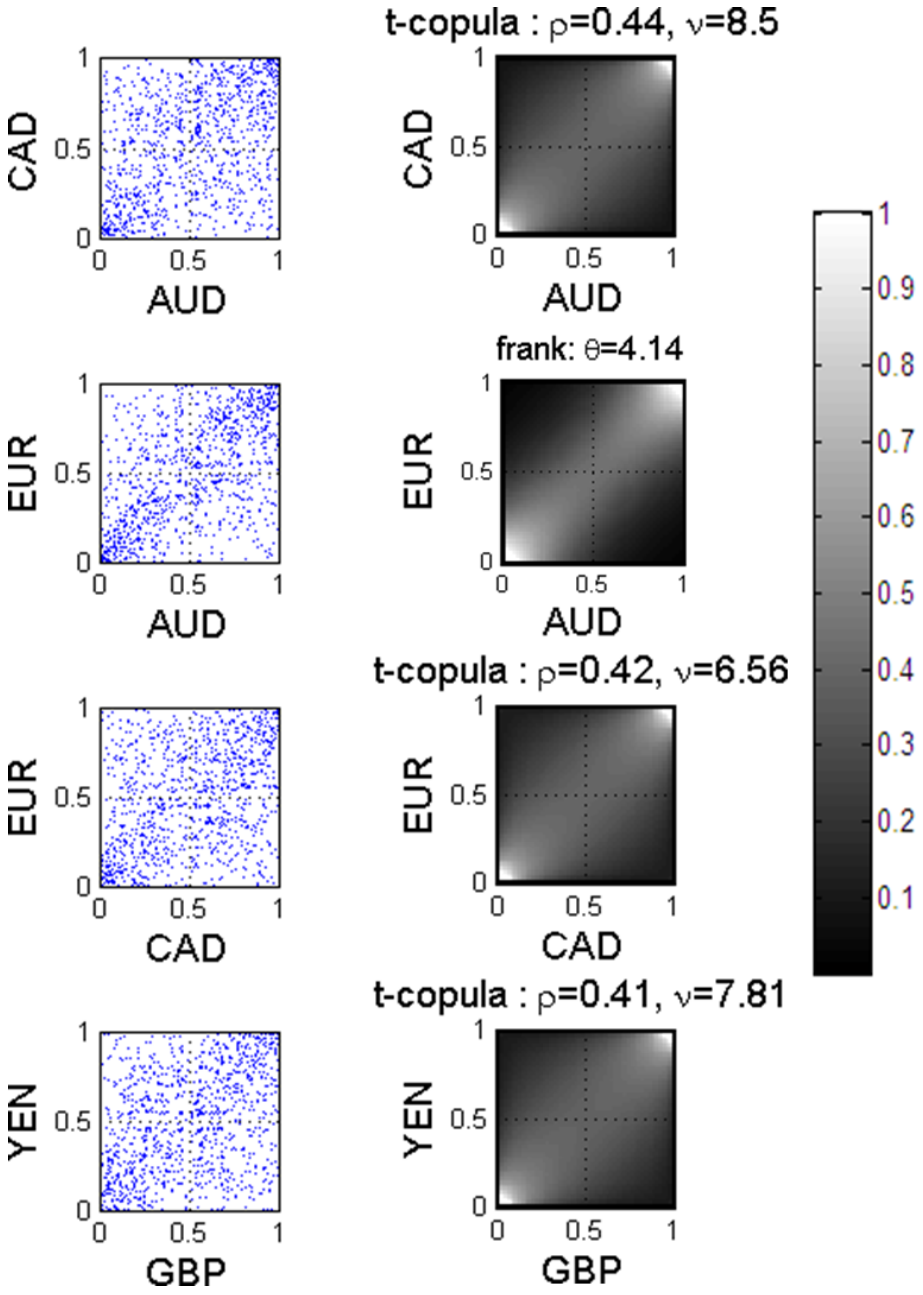

Figure 4.27: Bivariate binding copulae for currency rates. The best-fit binding copula functions for each of the currency rate pairs are shown on the right column along with the copula types. The corresponding scatter-plot, repeated from Figure 4.26 are shown on the left column for ease of comparison.

Moreover, we compared the parameter and likelihood values estimated via one-stage and two-stage MLE are not dramatically different. The AUD-USD and CAD-USD exchange rate series are used as an example. First, a set of parameters is estimated using the two-stage MLE as described in Sections 4.2.5 and 3.5.1. Then, using the two-stage MLE solution as the initial values, we estimated the entire set of parameters in a single step of MLE. The resulting parameters, along with the likelihood values achieved and run-time[11] required are listed in Table 4.6. The final likelihood values show the one-stage estimation method resulted in slight improvements over the two-stage estimation method at the expense of severe run-time increase.

[11]The run-time was recorded on a notebook computer with a Intel Core2 T7200 processor and 3GB of RAM. The estimation procedures were coded in Matlab (non-compiled). No other applications are running on the same computer during the experiment.

Table 4.6: One-stage vs. two-stage maximum likelihood estimation for bi-variate KCPs. The AUD-USD and EUR-USD exchange series are used as an example. Two set of parameters are estimated using the one-state and two-stage MLE method along with the associated likelihood values are tabulated below.

	Parameter	Description	Two-stage		One-stage	
			AUD	EUR	AUD	EUR
1.	κ	1/length-scale	0.0817	0.0652	0.0987	0.0532
2.	σ	average volatility level	3.59	2.12	2.50	2.61
3.	α	amplitude of volatility cycles	1.08	0.89	0.96	1.05
4.	ϕ	phase of volatility cycles	0.74	0.89	0.72	0.92
5.	λ	marginal skewness	-0.57	-0.65	-0.34	0.43
6.	ν_M	marginal DoF	3.42	3.57	5.12	6.39
7.	ν_C	copula DoF	6.88	7.88	6.01	6.23
8.	σ_n	noise density	0.012	0.029	0.041	0.018
9.	θ_f	frank copula parameter	4.14		4.32	
	$-\log \mathcal{L}(\theta)$	Maximized Likelihood Value	2561.3		2592.7	
		Run-time	2.64 mins		65.7 mins	

Now, we shift our focus to the temperature data set. Similar to the currency data set, there exists no natural ordering among the weather stations. We will again take advantage of the modularity of the KCPs and connect the univariate KCPs learned for each weather station with a binding copula. To illustrate a different binding method, instead of considering the pair-wise codependent structure, we group the weather stations by the states where they located and consider the high-dimensional codependent structure.

The Gaussian and Student's t-copula are used as a binding copula to describe the interdependencies among the temperature series. In this case the Gaussian RBF kernel (3.4) is used, with the weighted distance metric:

$$d(i,j)^2 = d_{\texttt{lat}}^2 + d_{\texttt{long}}^2 + w \cdot d_{\texttt{elev}}^2 \tag{4.2}$$

where $d(i,j)$ is the weighted distance between weather station i and j, d_x are distances along latitude, longitude, and elevation, and w is the weight on the distance along elevation.

Figure 4.28 shows the Gram matrices of binding copulae and pair-wise correlation matrices of the detrended data after transforming by the distribution induced by the univariate KCPs that was earlier ranked to be the best. Recall that for non-Gaussian random variables, the Gram matrix is not identical to the covariance matrix, but rather it dictates the behavior of the covariance. Nevertheless, the simple Student's t-copula with only geographical distance information was able to recover a substantial amount of interdependency on a visual basis.

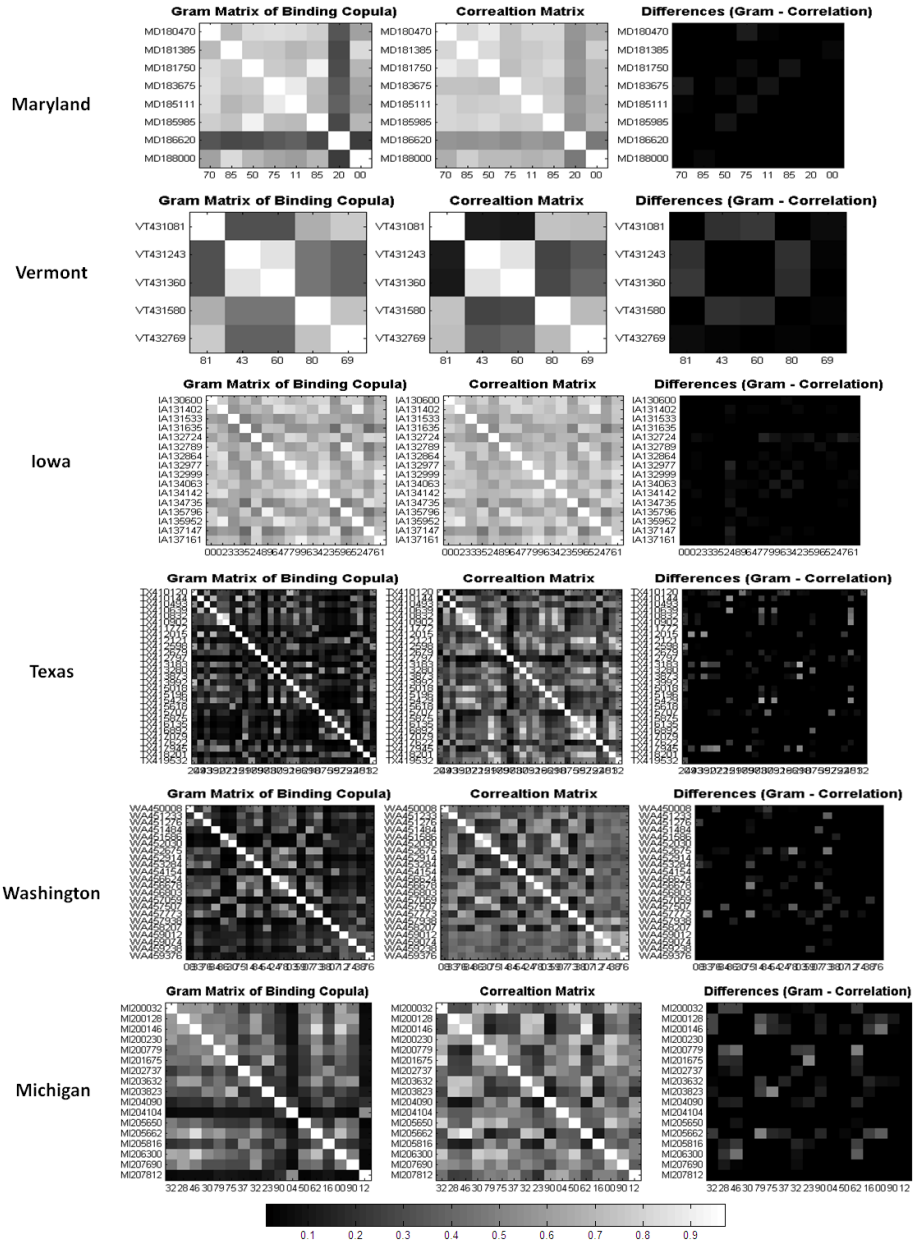

Figure 4.28: Sample multivariate KCP results from a few states. The Gram matrix (left) of the binding copula, pairwise correlation (middle), and the difference of the two matrices (right). The IDs of the weather stations are labeled on the vertical axes while only the last two digits of the IDs are shown on the horizontal axes.

4.3 A Note on Run-Time

In this section, we provide a comparison of run-times required by each model to perform learning on each data set. The run-times were recorded on a single node of a 64-node computing cluster[12]. Each node on the linux-based computing cluster is equipped with 12 GB of RAM and two dual-core AMD Opteron 2220 processors running at 2.8 GHz. All experiments are conducted based on codes written in Matlab 7.5 (R2007b).

4.3.1 Experiment Setup

The data set used for run-time measurements are identical to the data sets used in Section 4.2. Here we measure time required for learning model parameters using one training window (i.e. 100 data points) of the training data. Some of the Matlab code that we used in this experiment were generously made available by the authors as listed in Table 4.7. Note that the Matlab code by Rasmussen and Williams [85] is used for the GP even though the KCP codes can be readily used by using the Gaussian Processes. This is a deliberate choice to make the run-time results a more realistic reflection of practical performance as Rasmussen and Williams' codes are optimized for GPs. That is, the computation for the GPs in Rasmussen and Williams' codes only use matrix manipulations to compute the likelihood, the mean, and the covariance functions of the GPs (see Equations (2.5) and (2.2.1).). Whereas the likelihood function of the KCPs involves the explicit computation of density functions (see Equation (3.56)).

4.3.2 Measurements and Discussion

The run-time measurements for each data set and each model are tabulated in Table 4.8. Recall the figures represent the amount of time required for each model to learn the

[12]The computer cluster belongs to the information processing lab at the University of Toronto, a joint effort between Profs. B. Frey, F. Kschischang, and W. Yu. The author thank Prof. B. Frey for the access to the cluster.

Table 4.7: Matlab codes used for the experiment.

Model	Source of the codes used
GARCH	Matlab GARCH toolbox.
WGP	Matlab code By Snelson et. al. [100]
GCGM-KCP (GP)	Matlab code by Rasmussen and Williams [86]
GCSM-KCP	Matlab code by the author.
TCGM-KCP	Matlab code by the author.
TCSM-KCP	Matlab code by the author.

respective parameters for a single training window of 100 data points. The run-times are then averaged across the entire data series. In the case of the currency rates and the temperature data sets, the run-time is also averaged across data series within the data set.

Notice the GARCH(1,1) model is the fastest model. This is not surprising because of the model simplicity. On the other hand, the warped GP is the slowest model. This is in addition to the fact that the run-times (as well as goodness-of-fit as shown earlier in this chapter) recorded here are already significantly improved by using the maximum likelihood parameters learned with GPs as initial values thus eliminating the number of random-restarts required. However, the warped GPs do require the learning of a much larger number of parameters and numerically computing the derivative of the warping function in the likelihood function (Equation (2.8)) further adds to the computational burden. Moreover, focusing on the four types of KCPs (including GPs), the run-time grows linearly with the model complexity as expected. It is interesting to note that the GP is significantly ($\sim$ 2x) faster than the KCPs model. This is likely due to the GP code being optimized. Nevertheless, the run-times of all KCPs (including when the KCP reduces to the GP) are of the same order. The results may suggest that the main

Table 4.8: Summary of run-times. The following run-times were recorded for each data set and each model. Due to the large volume of data, the average run-times are reported for the currency rates (FX) and temperature data set. All measurements are presented in milliseconds, averaged over a single training window (i.e. 100 data points).

	Data Set			
Model	VIX	WTI	FX	Temperature
GARCH(1,1)	76.2	72.4	72.2	79.1
WGP	732.6	727.7	741.5	729.0
GCGM-KCP	141.6	165.6	170.8	166.6
GCSM-KCP	265.3	265.0	266.2	286.9
TCGM-KCP	303.2	271.6	278.5	293.0
TCSM-KCP	328.4	284.6	293.1	303.1

computation bottleneck remains to be the inversion of the covariance matrix, which is $\mathcal{O}(N^3)$ in the number of data points.

In conclusion, the KCPs model demands a modest run-time premium over the GARCH and GP model. However, the KCPs do provide superior predictive performance which may be necessary for some applications.

4.4 Summary of Results

In this chapter, the KCP model is applied to an array of real-life data sets, including the volatility index, crude oil future prices, foreign exchange rates, and temperature time-series. A list of popular time-series analysis models such as the GARCH(1,1), warped GP, and GP are also included for comparison.

First, we study the data sets as univariate series. To showcase the flexibility and versatility of the KCP model, different combinations of copula functions and marginal distributions are used. For fair comparisons, the same OU or heteroskedastic kernels are used for all models except the GARCH models. In all cases, the GCSM and TCSM KCPs have shown superior performance according to the BIC and the AD score over the competing models. However, the exact ranking of goodness-of-fit among models may occasionally be different when the in-sample and predictive metrics are used. The VIX data set has been the only exception where the GARCH(1,1) model performed better than the KCP models according to BIC. This is not surprising as the GARCH model is specifically designed to model phenomenon such as volatility clustering in time-series. Nevertheless, the GCSM and TCSM KCPs performed best as a predictive model for the VIX data set according to the AD score. A higher order GARCH model may improve performance, but the model complexity would increases linearly.

The warped GPs on the other hand, delivered improved PIT performance over GPs as was intended. Its BIC performance was severely penalized by the large number of parameters that were needed for the nonlinear warping function. However, the gain in performance also comes with a disproportionate increase in computation times. Furthermore, the absolute performance of the warped GPs in terms of BIC and AD scores did not live up to expectation in handling non-Gaussian data, showing that a meaningful performance gain must be achieved through a fundamental shift in paradigm such as what we have done in KCPs.

The superior performance of the GCSM and TCSM KCPs can be explained by their

ability to prescribe customized marginal distributions to a specific data set. It is interesting to note that the use of Student's t-copula did not improve the performance much over the use of Gaussian copula with the same choice of marginal distribution. In fact, the performance of KCPs with the Student's t-copula can be worse when ranked under BIC, as it is penalized by one extra parameter.

Finally, we shifted our focus to the codependent structure that exists among multiple series in the currency rates and temperature data sets. For the relatively small currency data set, we explored the codependent structures using bivariate binding copulae. As it turns out only four pairs of currency rates (AUD-CAD, AUD-EUR, CAD-EUR, and GBP-YEN) contained significant pairwise codependence. Further, we used six different copula functions including the Gaussian, Student's t, Frank, Gumbel, Clayton, and SJC copulae. Under the ranking according to the BIC, the Student's t performed best for all currency pairs except the AUD-EUR pair for which the Frank copula fit better. Note that when modeling multivariate data, the GPs and warped GPs are effectively confined to the use of Gaussian binding copula. In this study, we have shown that just as in choosing the type of marginal distributions, Gaussian is not always the best choice when choosing binding copula. The important message in this study is not which particular copula function fits the data set best, but the flexibility that the KCP model offers. With the use of a multivariate binding copula function, we further demonstrate the use of KCPs in a high-dimensional setting with marvelous results as evident by the close resemblance of the learned correlation structure to the real data.

Chapter 5

Conclusions and Future Directions

In this work I have created a flexible, versatile, and compact model for non-Gaussian and non-stationary time-series: Kernel-based Copula Processes. The (KCPs) take advantage of the unique properties of copula functions and the power and flexibility of kernel functions to cleanly separate the co-dependent structure of random variables from the marginal distribution, while capturing virtually any complex long-range codependency, all within a natural probabilistic framework.

5.1 Original Contributions

The contributions of this thesis are manifold and are summarized in turn below.

Univariate KCPs and new kernels design. We have proposed the first copula-based time-series model for directly modeling long range dependency and marginal behavior in a univariate time-series setting. Starting with elliptical copulae as the foundation for the univariate KCPs, not only did I exploit the unique properties of copula functions to allow flexible marginal behavior, but I also leveraged the multi-dimensional extension of elliptical copulae to capture the codependent structure in time. Further, I employed kernel functions with elliptical copulae – which proved to be central to the success of the KCP model. The kernel function not only introduced a sense of time to the model,

but it allowed additional dynamics (e.g. changes in variance) to be captured, making extensions to non-stationarity possible. I have also demonstrated a principled way to construct kernel functions using stochastic differential equations (SDEs) that is highly customized to the specific applications. Not only have I unlocked additional kernel functions to complement the library of kernel functions already known to the machine learning community, but the construction method via SDEs has also opened the opportunities for the mathematical finance community to transfer their SDEs to create kernel functions with more complex behavior.

Multivariate KCP extension. Recognizing the practical importance of analyzing multiple time series simultaneously, I built on the univariate KCPs framework to construct a multivariate extension. The most distinguishing feature of the multivariate KCPs is that each constituent time-series can assume vastly different characteristics and dynamics as each of them are modeled by an independent univariate KCP. The codependent structure among these time series are captured using a binding copula that is designed to tie the univariate predictive distributions together. The resulting model yields a principled framework with an efficient two-stage learning procedure.

Applications to real-life data sets. Moreover, I have demonstrated the practical usefulness of the KCPs through successful application to a wide array of real-life data sets that are challenging for conventional models because of their non-Gaussian and heteroskedastic characteristics. The KCPs produced high-quality and superior predictive distributions for univariate series predictions and uncovered complex codependencies among multiple time-series. Throughout the process, I have provided a systematic approach to analyzing data using the KCPs to guide the design choices regardless of the domain of application. I also applied the kernels that I have designed via SDEs and demonstrated their effectiveness in real-life applications.

Application of KCPs to classification problems. Finally I delved into the domain of classification problems. Remarkably, the KCP model lends itself seamlessly

to perform non-trivial classification tasks simply by choosing an appropriate marginal distribution. The KCPs perform extremely well on nonlinearly separable synthetic data while completely avoiding the complicated and computationally expensive procedures required by other methods such as Laplace approximation for the Gaussian processes classification [85].

5.2 Future Research Directions

While the KCP model has addressed and fulfilled many immediate needs of practical time-series analysis, it also raised many interesting future research directions and opportunities:

Change point detection. Often, financial time-series go through periods of high and low volatilities as the market goes through calm and distressed times. One common way to model such changes in market sentiments is the so-called *regime-switching* model, where one or more state variables such as the volatility is assumed to be in a set of discrete states or regimes. The same concept can be extended to model changes to different latent qualities of the price or economic indicator processes. The challenge of identifying regime changes or transitions is known as *change-points detection* [28] [5] [68]. In one possible approach, one could further invoke the conditional independence property under multiple regimes which states that given the position of a changepoint, the data before that changepoint is independent of the data after the changepoint. Given the $R+1$ regimes and positions of changepoints $l_1, \ldots, l_R$, a distinct KCP can be used to model each regime or segment of time-series, thus the joint probability of a time-series can be written as

$$\mathbb{P}\left(\{y_t\}_{t=1}^{N}\right) = \prod_{r=1}^{R} \int \mathbb{P}\left(\{y_t\}_{t=l_r}^{l_{r+1}} \middle| \theta_r\right) \mathbb{P}(\theta_r) d\theta_r \tag{5.1}$$

where $\mathbb{P}\left(\{y_t\}_{t=l_r}^{l_{r+1}} \middle| \theta_r\right)$ is a KCP with the parameters θ_r describing the r^{th} regime. Of course, the main challenges are the fact that the number of regimes R and the durations

$s_r = l_{r+1} - l_r$ are unknown and they should be modeled as latent variables. The overall framework can be treated in a Bayesian fashion by introducing proper priors for R and $\{s_r\}$,

$$\mathbb{P}\left(\{y_t\}_{t=1}^{N}\right) = \sum_{R}\sum_{s_r}\prod_{r=1}^{R}\int \mathbb{P}\left(\{y_t\}_{t=l_r}^{l_{r+1}}\middle|\, \theta_r, s_r\right)\mathbb{P}(\theta_r)d\theta_r\mathbb{P}(s_r|R)\mathbb{P}(R) \tag{5.2}$$

Furthermore, to be of practical use, such a model must detect the occurrence of new changepoints in real-time. Thus a new online learning algorithm must be developed simultaneously.

Dynamic KCPs with state-space models. Despite the modeling versatility and superior predictive power of KCPs as demonstrated in this work, current KCPs framework is not well suited for processes with no apparent codependency in their first moment. There are many time-series that are driven by physical processes that are more or less deterministic but unobservable or too complex to parameterize explicitly. Examples include (1) in speech recognition, each sound is made by a series of deliberate muscle movements and vocal cord vibrations; (2) in computer graphics and animations, human motion capture data only provides the positions of a relatively few markers that are driven by the movements of a large number of joint angles with complex kinematic and dynamical constraints.

The KCPs can be extended to include hidden dynamics by leveraging a state-space framework. One possible future research direction is to use KCPs to drive a latent dynamic process, while the latent states are linked to the observations via an emission probability such as the ones in hidden Markov-Models (HMMs). As such, the underlying physical process will be modeled by a nonlinear process with long-range memory. Furthermore, a multivariate KCP will be ideal to model complex underlying physical processes with a large number of states and diverse dynamics. This will provide an improvement of the vanilla Kalman filter as the dynamics described by KCPs are both nonlinear and possess long-memory. In this manner, the evolution of dynamics can be

completely data driven.

Option pricing engine. Pricing of contingent claims (e.g. option contracts) has been central to research in financial engineering since the seminal works of Black and Scholes [8] and Merton [73] in 1973. In an *arbitrage-free market*[1] with rational participants, they were able to construct a replicating portfolio of an option using only the underlying stock and cash. Shortly after, Harrison, Kreps, and Pliska [44] & [45] developed the *First Fundamental Theorem of Asset Pricing*, which states that a market is arbitrage-free if and only if there exists an *equivalent martingale measure* (EMM)[2] and that the EMM is unique only if the market is *complete*[3]. The price of any financial derivative is simply calculated as the expectation under the EMM.

As most real markets are incomplete, many risk-neutral measures exist and a choice must be made to select a risk-neutral measure for asset pricing. Gerber and Shiu [37] introduced a celebrated tool in actuarial science known as the Esscher Transform, as one of the possible choices for risk-neutral measure. Consider a random process $X(t)$ for $t > 0$. The Esscher Transform of probability density is defined as follows:

$$f^{\mathbb{S}}(x,t;h) = \frac{e^{hx} f^{\mathbb{P}}(x,t)}{M(h,t)} \tag{5.3}$$

where $\mathbb{S}$ and $\mathbb{P}$ denote the Esscher-transformed measure and real-world measure respectively, and h is a real number such that

$$M(h,t) = \int_{-\infty}^{\infty} e^{hx} f^{\mathbb{P}}(x,t)dx \tag{5.4}$$

[1] An arbitrage opportunity is one in which an agent can make a risk-less profit by making a number of self-financing trades (i.e. use the proceeds from some transactions to pay for the others to achieve a cash-flow neutral initial position) as allowed by the market. An everyday example would be the bank taking depositors' money at a return of 3% p.a. Then the bank lend the money to a mortgage borrower at a higher rate, say 6% p.a. The bank makes a risk-less profit of 3% while achieving a neutral initial cash position; assuming the mortgage borrower has no chance of defaulting of course.

[2] An equivalent martingale measure is an equivalent measure to the real world measure under which all tradeable assets grow at the risk-free rate.

[3] A market is complete if the value of any asset can be attained by a self-financing trading strategy.

exists. Note that $M(z,t) = \mathbb{E}^{\mathbb{P}}[e^{zX(t)}]$ is the moment generating function of $X(t)$. The parameter h is determined such that the transformed probability measure under which the expected discounted payout of the asset, S_t, is a martingale. That is, the Esscher-transformed measure $\mathbb{S}$ becomes a risk-neutral measure $\mathbb{Q}$ and

$$S_t \cdot e^{r(T-t)} = \mathbb{E}^{\mathbb{Q}}[S_T|\mathcal{F}_t] = \mathbb{E}^{\mathbb{P}}[S_T \cdot \frac{e^{hx}}{M(h,t)}|\mathcal{F}_t] \tag{5.5}$$

where $T > t \geq 0$ are indexes of time, r is the risk-free rate, and $\mathcal{F}_t$ denotes the filtration up to time t which the price process S_t is adapted to the *time-to-maturity* $(T-t)$ is assumed to be sufficiently small such that r can be considered a constant. It can be shown that there exists a unique $h^{\mathbb{Q}}$ to transform the real-work measure $\mathbb{P}$ to the risk-neutral measure $\mathbb{Q}$. Following the recipe in Gerber and Shiu [37], $h^{\mathbb{Q}}$ can be determined by solving the following equation:

$$e^r = \frac{M(1+h^{\mathbb{Q}},1)}{M(h^{\mathbb{Q}},1)} \tag{5.6}$$

KCPs fit naturally into the above framework. For instance, consider the pricing of a European call option of a single asset. A KCP can be used to model the price process of an asset or its daily log-return process just as described in Section 4.2. The posterior distribution of the asset price at maturity can be readily obtained under the real-world measure. The arbitrage-free price can be obtained after translating the posterior distribution to the risk-neutral measure via the Esscher transform.

The major advantage of applying KCPs to the Esscher transform option pricing framework is that the price process is no longer limited to processes with independent and stationary increments. Dependence can be introduced via the copula function.

Risk management. With the series of recent financial crisis, risk management is taking center stage in the financial industry and national financial policies alike. Better tools are desperately needed to help decision makers manage risk exposures of their firms and financial policies in the hope of avoiding bankruptcies.

One of the most widely used risk measure is Value-at-Risk (VaR), which specifies a threshold value such that the probability that the mark-to-market loss on the portfolio over a given time horizon exceeds this value (assuming normal markets and no trading in the portfolio) is a given probability level [54]. Mathematically, given some confidence level $\alpha \in (0, 1)$ the VaR of the portfolio at the confidence level α is given by the smallest number l such that the probability that the loss L exceeds l is not larger than $(1 - \alpha)$ [72]:

$$VaR_\alpha(L) = \inf\{l \in \mathbb{R} : \mathbb{P}(L > l) \leq (1 - \alpha)\} \tag{5.7}$$

Despite of all its nice properties, the notion of VaR has some shortcomings. A common complaint among academics is that VaR merely provides a lowest bound for losses without being able to distinguish between their degrees [89]. Further, VaR is not *subadditive*, that is the VaR of a combined portfolio can be larger than the sum of the VaRs of its components. The Conditional Value-at-Risk (CVaR) was developed to address some of these criticisms of VaR. It is defined as

$$CVaR_\alpha(L) = \mathbb{E}\left[L |\, L > VaR_\alpha(L)\right] \tag{5.8}$$

The definition of CVaR for general distributions is defined as the weighted average of VaR and the expected losses that are strictly greater than VaR. The CVaR is subadditive, and is thus more intuitive and conservative as a risk measure.

Given the flexibility of the KCPs and their superior predictive performance demonstrated for a wide array of financial time-series, a natural application of the KCPs is to model the loss distribution $\mathbb{P}(L)$, which can then used to compute $CVaR_\alpha(L)$.

The KCPs can also help in other areas of risk management, for example, the KCPs can be used as a Monte-Carlo simulation tool for generating scenarios. The procedure described in Section 3.5.3 can be used repeatedly to generate a large number of sample paths for simulation purposes.

Classification. Finally, the KCP model is readily extendable to tackle the classifi-

cation problem as shown in Section 3.9. However, we have just begun to reveal the full potential of the KCPs in this area. Many questions are still unanswered. For example, what is the best way to incorporate domain knowledge in the KCP model? Can the training data be a guide for selecting the copula and kernel functions? Is there another more suitable marginal distribution? One potential direction is to consider using a discrete multinomial distribution as the marginal distribution since the class labels are discrete. The computations of the KCP classifier will be sped up significantly. However, care must be taken in that case as the copula function that represents the joint distribution of the data is no longer a proper distribution function [36]. Multiple KCPs can be used to focus on different regions of the input space, creating a hierarchial classification framework or mixtures of KCPs.

Appendix A

The Theory of Copula

This chapter provides an *incomprehensive* introduction to the theory of copula to support the development of the rest of the proposal. For a comprehensive overview of the theory of copula, please consult Nelson [79].

A.1 What is a Copula?

Qualitatively, a copula is a function that links (couples) the univariate marginal distributions to the joint distribution, or

$$H(x_1, ..., x_N) = \mathbb{C}(F_1(x_1), ..., F_N(x_N)) \tag{A.1}$$

where $H(x_1, ..., x_N) \in [0, 1]$ is the joint cumulative distribution of the multivariate observations $X = \{X_i\} \in \mathcal{R}^N$ and $F_i(x)$ are the corresponding marginal culmulative distributions. The function $\mathbb{C}(u_1, ..., u_n) : [0, 1]^n \to [0, 1]$ is known as the *copula* function which links the joint distribution to its univariate marginals and thus encompasses the dependencies among variables.

By Skylar's theorem [1] [99], if the marginal distributions are continuous, then there exists an unique copula function, $\mathbb{C}$. Conversely, if $\mathbb{C}$ is a copula and the set of $\{F_i\}$ are

[1] More details can be found in Section A.4.

the set of univariate distribution functions, then H is the joint distribution function with margins $\{F_i\}$.

A.2 Why Use Copula?

There are numerous advantages of using copulae to model dependencies in multi-variate data. First, note the decomposition of the joint distribution into the dependency structure (copula) and the constituent marginal distributions. This separation allows each variate to be described by an entirely different distribution (e.g. Gaussian, Pareto, Gamma, etc.)

Further, unlike the correlation coefficient which measures co-variations up to the second order, copula functions capture the *complete* dependence structure. This feature is especially important when considering financial time series in the application of risk management as the events of interests are often a function of complex dependencies of many risk factors.

However, the drawbacks of using the copulae are that:

1. very few parametric copulae can be generalized beyond the bivariate case;

2. the same is true for copula model selection where most goodness-of-fit tests are devised for a bivariate copula and cannot be extended to higher dimensionality; and

3. intuitive interpretation of copula-parameter(s) is not always available.

A.3 Properties of Copula

A copula $\mathbb{C}(u_1, \ldots, u_N)$ is a function which maps $[0,1]^n$ to $[0,1]$. Further, for a function to be a copula, one must satisfy the following properties:

1. $\mathbb{C}(u_1, \ldots, u_N)$ is *grounded*. I.e. $\mathbb{C}(\cdot) = 0$ when any $u_i = 0$;

2. $\mathbb{C}(u_1, \ldots, u_N) = \mathbb{C}(u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_N)$ when any $u_i = 1$; and

3. $\mathbb{C}(u_1, \ldots, u_N)$ is *n-increasing*. I.e. For every $\mathbf{a}$ and $\mathbf{b}$ in $[0,1]^n$ and $\mathbf{a} \leq \mathbf{b}$, $V_{\mathbb{C}}([\mathbf{a}, \mathbf{b}]) \geq 0$, where $V_{\mathbb{C}}$ is the volume of a box bounded by the vertices defined by $\mathbf{a}$ and $\mathbf{b}$. For example, if $n = 2$, then $V_{\mathbb{C}}([\mathbf{a}, \mathbf{b}]) = \mathbb{C}(a_2, b_2) - \mathbb{C}(a_1, b_2) - \mathbb{C}(a_2, b_1) + \mathbb{C}(a_1, b_1)$.

A.4 Sklar's Theorem

Let H be an n-dimensional distribution function with margins $F_1, \ldots, F_n$. There exists an n-copula $\mathbb{C}$ such that for all $\mathbf{x}$ in $\mathbf{R}^n$,

$$H(x_1, ..., x_N) = \mathbb{C}(F_1(x_1), ..., F_N(x_N)) \tag{A.2}$$

If all the margins are continuous, then $\mathbb{C}$ is unique. Conversely, if $\mathbb{C}$ is an n-copula and $\{F_i\}$ are distribution functions, then the function H defined above is an n-dimensional joint distribution of $\mathbf{x}$.

A.5 Families of Copula

This section provides an overview of the most commonly used families of parametric copulae. For a comprehensive repertoire, please consult [79].

Most parametric copulae have well-defined properties in the bivariate case but do not generalize to the high dimensional case. Here, copula functions with high-dimensional extension are introduced.

A.5.1 Elliptical Distributions and copulae

A multivariate elliptically distributed random variable $\mathbf{x}$ can be considered as an affine transformation of a standardized spherically distributed variable $\mathbf{y}$ (i.e. $\mathbf{y} \sim \mathcal{S}(\Psi)$):

$$\mathbf{x} \stackrel{d}{=} \mu + \mathbf{A}^T\mathbf{y} \tag{A.3}$$

where μ and $\mathbf{\Sigma} = \mathbf{A}^T\mathbf{A}$ are the location vector and covariance matrix and Ψ is the generator function of the spherical distribution. An example of a spherical distribution is $\mathcal{N}(0, \mathbf{I})$.

An elliptical copula [31] bears the following form:

$$\mathbb{C}(u_1, \ldots, u_n) = F_\Sigma(h_1(u_1), \ldots, h_n(u_n)) \tag{A.4}$$

where F_Σ is a n-dimensional zero-mean elliptical distribution with a $n \times n$ correlation matrix Σ and $h_i(\cdot) : [0, 1] \to (+\infty, -\infty)$, $i = 1, \ldots, n$ are continuous and strictly monotonic increasing functions, typically an inverse cummulative distribution function. A more common form of elliptical copula function constrains the h_i and F_Σ to be in the same elliptical class such as Gaussian or Student's t copula with the following form:

$$\mathbb{C}(u_1, \ldots, u_n) = F_\Sigma(F_1^{-1}(u_1), \ldots, F_n^{-1}(u_n)) \tag{A.5}$$

where $F_i^{-1}(u_i)$ are standardized univariate elliptical distributions of the same type as F_Σ. Other examples of elliptical distributions includes Cauchy and Pearson VII. Please consult Fang *et. al.* [27] for more details.

A.5.2 Archimedean copulae

The Archimedean copula is a very broad and compact class of parametric copulae which also has an n-dimensional extension.

In general, the Archimedean copulae are controlled by a single-parameter θ and have the following form:

$$\mathbb{C}(u_1, \ldots, u_n) = \varphi_\theta^{-1}(\Sigma_{i=1}^n \varphi_\theta(u_i)), \tag{A.6}$$

Table A.1: Generator functions for Archimedean copulae.

Copula Family	Generator $\varphi_\theta(u)$	Conditions
Independent	$-\ln(u)$	
Frank	$\ln\left(\frac{e^{\theta u}-1}{e^{\theta}-1}\right)$	
Clayton	$u^{\theta}-1$	$\theta \leq 1$
Gumbel	$[-\ln(u)]^{\theta}$	

where φ_θ is known as the *generator* and is a continuous, strictly decreasing function mapping [0,1] to [0,∞]. The Archimedean copulae encompass many well-known families of copula functions such as the independent, Frank, Clayton, and Gumbel. The corresponding generator functions are listed in Table A.1.

For a comprehensive list of generator function, please consult Table 4.1 of Nelson [79].

A.6 Dependency Measure

In general, linear correlation measures such as the Pearson's correlation coefficient is not sufficient to capture the true dependencies of two random variables. Two dependency measures that are commonly used with copulae are introduced here: Spearman's Rho and Kendall's Tau. However, these dependency measures are only limited to the bivariate case.

First, define a pair of random variables X and Y and the corresponding set of observations $\{X_i, Y_i\}$. Construct the ordered set $\{R_i, S_i\}$, where R_i and S_i are the normalized rank of X_i and Y_i in ascending order, respectively.

A.6.1 Spearman's Rho

The Spearman's Rho ρ_n is defined as follows:

$$\rho_n = \frac{\sum_{i=1}^{n}(R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum_{i=1}^{n}(R_i - \overline{R})^2 \sum_{i=1}^{n}(S_i - \overline{S})^2}} \tag{A.7}$$

where $\rho_n \in [-1, 1]$ and

$$\overline{R} = \frac{1}{n}\sum_{i=1}^{n} R_i = \frac{n+1}{2} = \frac{1}{n}\sum_{i=1}^{n} S_i = \overline{S} \tag{A.8}$$

Like the Pearson's correlation coefficient r_n, the Spearman's Rho ρ_n is identically 0 when X and Y are independent. However, the Spearman's Rho is theoretically superior to Pearson's correlation coefficient [24][35] in that $\mathbb{E}(r_n) = \pm 1$ if and only if X and Y are linear functions of one another, while $\mathbb{E}(\rho_n) = \pm 1$ as long as X and Y are functions of each other, thus less restrictive. Further, ρ_n is always well-defined, whereas the r_n does not exists for some heavy-tailed distributions such as the Cauchy distribution.

As an interesting connection to copulae, the asymptotically unbiased estimator of ρ_n is

$$\rho = 12 \int \mathbb{C}(u, v) dv du - 3. \tag{A.9}$$

A.6.2 Kendall's Tau

The empirical version of Kendall's tau is defined as follows:

$$\tau_n = \frac{P_n - Q_n}{C_2^n} \tag{A.10}$$

where P_n and Q_n are the number of *concordant* and *discordant* pairs, respectively. Two pairs $(X_i, Y_i), (X_j, Y_j)$ are said to be concordant when $(X_i - X_j)(Y_i - Y_j) > 0$, and discordant when $(X_i - X_j)(Y_i - Y_j) < 0$.

As for Spearman's rho, the theoretical value of Kendall's tau is a monotonic increasing function of the real parameter θ whenever a family $(\mathbb{C}_\theta)$ of copulae is ordered by positive quadrant dependence.

The asymptotically unbiased estimator of τ_n is given by

$$\tau = 4 \int \mathbb{C}(u, v) d\mathbb{C}(u, v) - 1. \tag{A.11}$$

Appendix B

Likelihood Ratio Test

The likelihood ratio test is used compare the goodness-of-fit between two models:

$$\lambda_{p,q}(\mathcal{D}) = 2\ln\frac{\mathcal{L}(\widehat{\theta_p}|\mathcal{D})}{\mathcal{L}(\widehat{\theta_q}|\mathcal{D})} \tag{B.1}$$

where the $\mathcal{L}(\widehat{\theta_p}|\mathcal{D})$ and $\mathcal{L}(\widehat{\theta_q}|\mathcal{D})$ are the likelihood functions of the models $\mathcal{M}_p$ and $\mathcal{M}_q$ evaluated on data set $\mathcal{D}$. When the likelihood functions are evaluated at the parameter sets $\widehat{\theta_p}$ and $\widehat{\theta_q}$, the likelihood functions are maximized. The null hypothesis $\mathcal{H}_0$ is that model $\mathcal{M}_p$ is a better fit to the data than model $\mathcal{M}_q$, and the alternate hypothesis $\mathcal{H}_a$ is that $\mathcal{M}_q$ is a better fit to the data than model $\mathcal{M}_p$ [14].

Formally, the probability distribution of the test statistic can be approximated by a chi-squared distribution, with degrees of freedom Δ which equals to the difference in the number of parameters.

The following table shows the critical values c at different significance level α for the chi-squared distribution. These critical values are used throughout this work in determining the statistical significance when comparing the goodness-of-fit between two competing models. If other critical values c are needed at different significance levels, it can be calculated as follows:

$$c = \mathcal{F}^{-1}_{\mathcal{X}^2}(1-\alpha;\Delta) \tag{B.2}$$

where $\mathcal{F}^{-1}_{\mathcal{X}^2}(\cdot)$ is the inverse CDF of the chi-square distribution with degree of freedom Δ.

Table B.1: Critical values for the likelihood ratio test at various significance levels.

Significance Levels α			
Δ	0.10	0.05	0.01
1	2.71	3.84	6.64
2	4.60	5.99	9.21
3	6.25	7.82	11.34
4	7.78	9.49	13.28
5	9.24	11.07	15.09
6	10.64	12.59	16.81
7	12.02	14.07	18.48
8	13.36	15.51	20.09
9	14.68	16.92	21.67
10	15.99	18.31	23.21

Bibliography

[1] J. Aitchison and I. R. Dunsmore. *Statistical Prediction Analysis*. Cambridge University Press, Cambridge, U.K., 1975.

[2] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, pages 267–281, 1973.

[3] E. Alvarado, D. V. Sandberg, and G. Stewart. Modeling large forest fires as extreme events. *Northwest Science*, 72:66–75, 1998.

[4] T. W. Anderson and D. A. Darling. A test of goodness of fit. *The Journal of American Statistical Association*, 49(268):765–769, 1954.

[5] D. Barry and J. Hartigan. A Bayesian analysis for change point detection. *Journal of the American Statistics Association*, 88:309–319, 1993.

[6] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer-Verlag New York, Inc., New York, NY, USA, 1993.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, USA, 2006.

[8] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, pages 637–659, 1973.

[9] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986.

[10] W. Breymann, A. Dias, and P. Embrechts. Dependence structures for multivariate high-frequency data in finance. *Quantitative Finance*, 1(3):1–14, 2003.

[11] W. Breymann, Alexandra Dias, and P. Embrechts. Dependence structures for multivariate high-frequency data in finance. *Quantitative Finance*, 3:1–14, 2003.

[12] M. Carney and P. Cunningham. Evaluating density forecast models. TCD-CS-2006-21. Database, Trinity College Dublin, Department of Computer Science, 2006.

[13] P. Carr, D. B. Madan, and R. H. Smith. Option valuation using the fast fourier transform. *Journal of Computational Finance*, 2:61–73, 1999.

[14] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Press, 2001.

[15] CBOE. The cboe volatility index - vix (white paper), 2008. http://www.cboe.com/micro/vix/vixwhite.pdf.

[16] P. E. Ceruzzi. Sextant, apollo guidance and navigation system, 2000. http://www.ion.org/museum/.

[17] U. Cherubini, E. Luciano, and W. Vecchiato. *Copula Methods in Finance*. John Wiley & Sons, Ltd., West Sussex, U.K., 2004.

[18] OANDA Corporation. Historic foreign exchange rates database., 2010. http://www.oanda.com.

[19] W.F. Darsow, B. Nguyen, and E.T. Olsen. Copulas and markov processes. *Illinois Journal of Mathematics*, pages 600–642, 1992.

[20] P. Deheuvels. Caractrisation complte des lois extrmes multivaries et de la convergence des types extremes. *Publ Inst Statist Univ Paris*, 23:1–37, 1978.

[21] A. Dias and P. Embrechts. Dynamic copula models for multivariate high-frequency data in finance. *Pre-print*, 2004.

[22] F. X. Diebold, T. A. Gunther, and A. S. Tay. Evaluating density forecasts with applications to financial risk management. *International Economic Review*, 39(4):863–883, 1998.

[23] A. S. S. Dorvlo. Estimating wind speed distribution. *Energy Conversion and Management*, 43(17):2311–2318, 2002.

[24] P. Embrechts, A. J. McNeil, and D. Straumann. Correlation and dependence in risk management: Properties and pitfalls. *Risk management: Value at risk and beyond (Cambridge, 1998)*, page 176V223, 2002.

[25] R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of variance of united kingdom inflation. *Econometrica*, 50:987–1008, 1982.

[26] E. F. Fama. The behavior of stock-market prices. *Journal of Business*, 38(1):34–105, 1965.

[27] K-T Fang, S. Kotz, and K-W Ng. *Symmetric Multivariate and Related Distributions*. Chapman and Hall, 1990.

[28] P. Fearnhead. Exact and efficient Bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16:203–213, 2006.

[29] E. D. Feigelson and G. J. Babu. Statistical challenges in modern astronomy. In *Proceedings of the Statistical problems in Particle Physics, Astrophysics and Cosmology (PHYSTAT2003)*, September 2003.

[30] J-D Fermanian and M. Wegkamp. Time dependent copulas, working paper, 2004.

[31] G. Frahm, M. Junker, and A. Szimayer. Elliptical copulas: applicability and limitations. *Statistics and Probability Letters*, 3(63):275–286, 2003.

[32] M. Fréchet. Sur les tableaux de corrlation dont les marges sont donnes. *Ann Univ Lyon Sect A*, 9:53–77, 1951.

[33] J. Galambos. *The Asymptotic Theory of Extreme Order Statistics*. Wiley, New York, New York, USA, 1978.

[34] A. Gelb. *Applied Optimal Estimation*. The MIT Press., 1974.

[35] C. Genest and A.C. Favre. Everything you always wanted to know about copula modeling but were afraid to ask. *The Journal of Hydrologic Engineering*, 12:347–368, 2007.

[36] C. Genest and J. Neslehova. A primer on copulas for count data. *Astin Bulletin*, 2(37):475–515, 2007.

[37] H. U. Gerber and S. W. Shiu. Option pricing by esscher transform. *Transactions of the Society of Actuaries*, pages 99–191, 1994.

[38] Z. Ghahramani and G. Hinton. Parameter estimation for linear dynamical systems. Technical Report Technical Report CRG-TR-96-2, University of Toronto, 1996.

[39] J. D. Gibbons and S. Chakraborti. *Nonparametric Statistical Inference*. CRC Press, 2003.

[40] P. W. Goldberg, C. K. I. Williams, and C. M. Bishop. *Regression with Input-Dependent Noise*. MIT Press, 1998.

[41] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, 2001.

[42] J. D. Hamilton. *Time-Series Analysis*. Princeton University Press, Princeton, New Jersey, USA, 1994.

[43] B. E. Hansen. Autoregressive conditional density estimation. *International Economic Review*, 35(3):705–730, 1994.

[44] J. M. Harrison and D. M. Kreps. Martingales and arbitrage in multiperiod securities markets. *Journal of Economic Theory*, pages 381–408, 1979.

[45] J. M. Harrison and S. R. Pliska. Martingales and stochastic integrals in the theory of continuous trading. *Stochastic Processes and Their Applications*, pages 215–280, 1981.

[46] H. Hassani. Singular spectrum analysis: Methodology and comparison. *Journal Data Science*, 5:239–257, 2007.

[47] F. Hayashi. *Econometrics*. Princeton University Press, Princeton, New Jersey, USA, 2000.

[48] W. Hoeffing. Masstabinvariante korrelationstheorie [reprinted as: Scale-invariant correlation theory.]. *Schriften des Matematischen Instituts und des Instituts fr Angewandte Mathematik der Universitt*, pages 179–233, 1940.

[49] W. Hoeffing. Masstabinvariante korrelationsmasse fr diskontinuierliche verteilungen [reprinted as: Scale-invariant correlation measures for discontinuous distributions.]. *Arkiv fr matematischen Wirtschaften und Sozialforschung*, pages 49–70, 1941.

[50] J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, New Jersey, USA, fifth edition, 2003.

[51] F. Jelinek. Statistical methods for speech recognition. In *Proceedings of the 37th Annual Meeting of the ACL*, 1997.

[52] H. Joe and J.J. Xu. The estimation method of inference functions for margins for multivariate models. Technical Report 166, Department of Statistics, University of British Columbia, September 1996.

[53] M. I. Jordon. *An Introduction to Probabilistic Graphical Models*. Unpublished. University of California, Berkeley, CA, USA, 2003.

[54] P. Jorion. *Value at Risk: The New Benchmark for Managing Financial Risk*. McGraw-Hill, New York, NY, 2006.

[55] C.N. Williams Jr., M.J. Menne, R.S. Vose, and D.R. Easterling. United states historical climatology network daily temperature, precipitation, and snow data. Database, The Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, U.S. Department of Energy, 2006.

[56] J. N. Juang. *Applied System Identification*. PTR Prentice Hall, Inc., New Jersey, USA, 1994.

[57] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. Am. Soc. Mech. Eng., Series D, Journal of Basic Engineering*, (82):35–45, 1960.

[58] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

[59] R. W. Katz. On some criteria for estimating the order of a markov chain. *Technometrics*, 23:243–249, 1981.

[60] E. Keogh, S. Lonardi, and B. Chiu. Finding surpprising patterns in a time series database in linear time and space. In *Proceedings of the ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD2002)*, Edmonton, Alberta, Canada, July 2002.

[61] G. Kimeldorf and A. Sampson. Uniform representations of bivariate distributios. *Comm Statist A - Theory Methods*, 4:617–627, 1975.

[62] V. V. Kleiza. Calculation of inverse functions by the monto carlo method. *Lithuanian Mathematical Journal*, 16(2):117–120, 1976.

[63] B. ksendal. *Stochastic Differential Equations: An Introduction with Applications*. 6th edition. Springer-Verlag New York, Inc., Cambridge, U.K., 2003.

[64] Rabiner L and Juang B. *Fundamentals of Speech Recognition*. Prentice Hall PTR, 1993.

[65] A. Lendasse, E. Oja, O. Simula, and M. Verleysen. Time series prediction competition: The CATS benchmark. In *International Joint Conference on Neural Networks*, Budapest, Hungary, 2004.

[66] L. J. Levy. The kalman filter: Navigation's integration workhorse. *GSP World*, 1997.

[67] H. Lutkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, New York, New York, USA, 2006.

[68] M. Maheu and T. McCurdy. How useful are historictal data for forecasting the long-run equity return distribution? *Journal of Business and Economic Statistics*, 27(1):95–112, 2009.

[69] B. Mandelbrot. The variation of certain speculative prices. *Journal of Business*, 26:394–419, 1963.

[70] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[71] G. Marsaglia and J. C. W. Marsaglia. Evaluating the anderson-darling distribution. *The Journal of Statistical Software*, 9(2).

[72] A. McNeil, R. Frey, and P. Embrechts. *Quantitative Risk Management: Concepts Techniques and Tools*. Princeton University Press, Princeton, New York, 2005.

[73] R. C. Merton. The theory of rational option pricing. *Bell Journal of Economics and Management Science*, pages 141–183, 1973.

[74] Distance Metrics. http://www.improvedoutcomes.com/.

[75] T.P. Minka. A family of algorithms for approximate Bayesian inference. (ph.d. thesis), 2001.

[76] K. Mosegaard and A. Tarantola. Monte carlo sampling of solutions to inverse problems. *Journal of Geophysical Research*, 100:12,431–12,447, 1995.

[77] R. M. Neal. *Bayesian Learning for Neural Networks.* Lecture Notes in Statistics. Springer, New York, 1996.

[78] R. M. Neal. Monte-Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report Technical Report 9702, University of Toronto, 1997.

[79] R. B. Nelsen. *An Introduction to Copulas.* Springer Series in Statistics. Springer-Verlag New York, Inc., New York, NY, USA, 2006.

[80] A. Oppenheim and R. Schafer. *Discrete-Time Signal Processing.* Springer Series in Signal Processing. Springer-Verlag New York, Inc., New York, NY, USA, 2009.

[81] A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, New York, USA, 2002.

[82] A. J. Patton. *Applications of Copula Theory in Financial Econometrics.* PhD thesis, University of California, San Diego, 2002.

[83] A. J. Patton. Modelling asymmetric exchange rate dependence. *International Economic Review*, 47(2):527–556, 2006.

[84] K. Pearson. On a method of determining whether a sample of size n supposed to have been drawn from a parent population having a known probability integral has probably been drawn at random. *Biometrika*, 25:379–410, 1933.

[85] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, USA, 2006.

[86] C. E. Rasmussen and C. K. I. Williams. Gaussian processes matlab code., 2006. http://www.Gaussianprocess.org/gpml/code/matlab/doc/.

[87] H. E. Rauch. Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 8.

[88] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimats of linear dynamic systems. *American Institute of Aeronautics and Astronautics (AIAA)*, 3(8):1445–1450, 1965.

[89] R.T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26(7), 2002.

[90] E. Rogge and P. Schonbucher. Modelling dynamic portfolio credit risk (working paper). Technical report, ETH Zurich, 2003.

[91] S. Roweis. Lecture notes from machine learning CSC2515, university of toronto, 2005. http://www.cs.toronto.edu/ roweis/.

[92] S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 2(11):305–345, 1999.

[93] G. Salvadori, N. T. Kottegoda, R. Rosso, and C. De Michele. *Extremes in nature: an approach using Copulas.* Springer, New York, USA, 2007.

[94] B. Scholkopf and A. Smola. *Learning wiht Kernels.* MIT Press, Cambridge, MA, USA, 2002.

[95] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[96] R. Shibata. Selection of the order of an autoregressive model by akaike's information criterion. *Biometrika*, 63:117–126, 1976.

[97] S. E. Shreve. *Stochastic Calculus for Finance II: Continuous-Time Models*. Springer Finance Textbook. Springer-Verlag New York, Inc., New York, NY, USA, 2000.

[98] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

[99] A. Sklar. Fonctions de repartition a n dimensions et leurs marges. *Publications de l'Institut de Statistique de L'Universite de Paris*, pages 229–231, 1959.

[100] E. Snelson. Wapred Gaussian processes matlab code., 2004. http://www.gatsby.ucl.ac.uk/ snelson/.

[101] E. Snelson, C.E. Rasmussen, and Z. Ghahramani. Wapred Gaussian processes. In *Proceedings of the Neural Information Processing Systems (NIPS2004)*, Vancouver, British Columbia, Canada, December 2004.

[102] P. X-K Song. Multivariate dispersion models generated from Gaussian copula. *Scandinavian Journal of Statistics*, 27(2):305–320, 2000.

[103] P. X.-K. Song, Y. Fan, and J.D. Kalbfleisch. Maximization by parts in likelihood inference. Working Papers 0319, Department of Economics, Vanderbilt University, September 2003.

[104] C.W. Sul, S.K. Jung, and K.Y. Wohn. Synthesis of human motion using kalman filter. *Modelling and Motion Capture Techniques for Virtual Environments*, 1537:100–112, 1998.

[105] D. Totouom and M. Armstrong. Dynamic copula processes: A new way of modelling cdo tranches. In *The 5th Annual Advances in Econometrics, Louisiana State University*, 2005.

[106] A. Wang. Shazam - an industrial-strength audio search algorithm. In *Proceedings of the International Society for Music Information Retrieval (ISMIR2003)*, October 2003.

[107] J. Wang, D. Fleet, and A. Hertzmann. Gaussian process dynamical models. In *Proceedings of the Neural Information Processing Systems (NIPS2005)*, Vancouver, British Columbia, Canada, December 2005.

[108] C. Wells. *The Kalman Filter in Finance.* Springer Netherlands., Netherlands, 1996.

[109] R. Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach.* The Wiley Finance Series. Wiley, 2006.

[110] C.K.I. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–51, 1986.