University of the Aegean

Department of Mathematics

Hidden Markov Models and their applications in Finance

Anastasios Petropoulos

A thesis submitted for the fulfillment

of the requirements for the degree of

Doctor of Philosophy

June 2015

Members		**Supervisor**
Chatzis Sotirios	Giannakopoulos Thanasis	Xanthopoulos Stelios

Abstract

Hidden Markov Models, usually referred to as HMMs, are one of the most successful concepts in statistical modeling conceived and analyzed in the last 40 years. They belong to the stochastic mixture models family and have been broadly implemented in numerous sectors to address the problem of data model fitting and forecasting. Their structure usually is comprised by an observed sequence which is conditioned on an underlying hidden (unobserved) process. This way HMMs provide flexibility to address various complicated problems and can be implemented for modeling univariate and multivariate financial time series. Moreover, based on current literature, economic variables exhibit patterns dependent on different economic regimes which can be successfully captured by HMMs. Their parsimonious structure and attractive properties along with the existence of efficient algorithms for their estimation were the main drivers for the selection of HMM as the main topic of this thesis. Consequently, in this thesis we thoroughly investigate HMMs and their capabilities to simulate financial systems. The contribution of this study is threefold: First we perform an extensive review of HMM theory and applications. Our aim is to summarize the most significant applications of HMM with special focus in the field of finance. We offer a thorough and compact summary of the uses and the results of HMM in the last 40 years. Secondly, we extend the framework of HMMs by proposing a theoretical variation, injecting greater flexibility in their structure. Based on bibliography, in many real-world scenarios the modeled data entail temporal dynamics the patterns of which change over time. We address this problem by proposing a novel HMM formulation, treating temporal dependencies as latent variables over which inference is performed. Specifically, we introduce a hierarchical graphical model comprising two hidden layers: on the first layer, we postulate a chain of latent observation-emitting states, the temporal dependencies between which may change over time; on the second layer, we postulate a latent first-order Markov chain modeling the evolution of temporal dynamics (dependence jumps) pertaining to the first-layer latent process. As a result of this construction, our method allows for effectively modeling non-homogeneous observed financial data. Finally in the third part of this thesis we investigate the HMM efficiency in the problem of corporate credit scoring. We propose a novel corporate credit rating system based on Student's-t hidden Markov models (SHMMs). Corporate credit scoring is widely used by financial institutions for portfolio risk management, and for pricing financial products designed for corporations. In addition, from a regulatory perspective, internal rating models are commonly used for establishing a more risk-sensitive capital adequacy framework for financial institutions. We evaluate our method against other state of the art statistical techniques like Neural Networks, SVM, and logistic regression and conclude that SHMM offer significant improved forecasting capabilities.

Keywords: Corporate credit rating, hidden Markov model, Student's-t distribution, expectation-maximization, Basel framework, statistical machine learning, temporal dynamics, hidden Markov models, expectation-maximization, variable order, dependence jumps, volatility modeling.

«11:15, restate my assumptions: 1. Mathematics is the language of nature. 2. Everything around us can be represented and understood through numbers. 3. If you graph these numbers, patterns emerge. Therefore: There are patterns everywhere in nature. » - Movie "Pi" (1998)

This thesis is dedicated to my wife and my two beautiful daughters

Thank you for your patience, support, love and encouragement

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr Xanthopoulos, for his support, continued motivation, enthusiasm and most especially, his long-lasting patience. I would like to thank Dr. Chatzis for his invaluable help on the two research projects that became part of this thesis and for his insightful ideas and suggestions on some aspects of my academic research. Additionally, I would like to thank Dr Tsimikas and Dr Yannakopoulos for giving me the opportunity for this research work and for their guidance throughout these years.

Table of Contents

List of Tables

List of Figures

Acronyms List

Acronym	**Definition**
HMM	Hidden Markov Model
GHMM	Gaussian Mixture Hidden Markov Model
SHMM	Student-t mixtures Hidden Markov Model
EM	Expectation Maximization Algorithm
MLE	Maximum Likelihood Estimation
MWL	Maximum Weighted Likelihood
HSMM	Hidden Semi-MarkovModel
MSE	Mean Squared Error
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
TPM	Transition probability matrix
GSHMM	Gaussian Student-t HMM
SME	Small, Medium Enterprises

Symbols

T = Number of observation

L = Likelihood

l = the log likelihood

N = number of hidden states

M = the number of different values of the observed variable of an HMM in the discrete case or the number of components in a Gaussian Mixture distribution

S = (s_1 , ..., s_T) the group of possible hidden states

O = (o_1 , ..., o_T) an observation sequence

Q = (q_0 , ..., q_T)a sequence of hidden states of lengthTwith values from S

A = {α_{11} , α_{12}, ..., α_{NN}} the transition matrix of the hidden Markov chain

B = { $b_{q_i}(o_i)$} the matrix of the probability density of the observed sequence for every hidden states.

The definition of matrix B depends on whether the HMM analyzed is continuous or discrete

Π = {π_1 , ..., π_N} the initial probability distribution

λ = {A,B,π} the group of parameters that characterize an HMM

Chapter 1 Introduction

1.1 About this Thesis

A significant amount of academic research performed in finance nowadays aims to predict the evolution of significant indicators of the economy and asset prices. The basic tool for the analysis of financial time series is the science of statistics. The constant changes of the economic conditions, the volatility in prices, the business cycle, the monetary policy, the evolution of interest rates and the political situation by country are some of the main drivers that increase the complexity of financial time series. This in turn leads to the enrichment of the statistical theory with new, synthetic and complicated models for increasing their efficiency in forecasting of economic variables. Each of these statistical methods presents advantages and disadvantages and usually there is not a one fits all remedy. Furthermore, the introduction of electronic and algorithmic trading also gave a boost in the need for improved forecasting of financial assets prices. In recent years HMM have been successfully implemented to address the problem of forecasting financial times series, exhibiting positive results against other state of the art techniques. In this thesis HMM are explored in depth, their applications are investigated and their efficiency is expanded by proposing a theoretical extension of their structure, to address better the analysis of financial data like shares, exchange rates, the oil market and gold. Moreover this study explores the predictive ability of HMM in assessing the credit quality of companies. The research belongs to the interdisciplinary field of computational financial. In a similar way to traditional econometrics, we build and implement models for analyzing and forecasting financial time series. However our approach differs in that the models used in the context of this analysis is not based on regression techniques (ARIMA, GARCH, ARCH), but instead belong to the field of machine learning and pattern recognition.

This chapter describes the basic concepts of financial time series, the difficulties of accurate economic forecasts, the motivation and reasons for selecting HMM as the main topic for our research and the contribution of this thesis in the field of computational finance. The organization of the remaining chapters of the thesis is also described in the last section of this chapter.

1.2 Financial Time Series and forecasting

Since the 1960's numerous studies have been implemented involving the analysis and prediction of the financial markets and the significant macroeconomic indicators. Many analysts and economists presented articles, thesis, with the aim of better modeling economic variables and achieve more accurate predictions for risk management purposes and better investment selections. Established model frameworks are summarized into four main categories of economic analysis: the fundamental analysis, technical analysis, econometric time series analysis and recently machine learning methods.

Fundamental analysis refers to the examination of the underlying forces that affect the wellbeing of the economy, industry sectors and individual companies. For example, to forecast future price of a stock, fundamental analysis combines economic situation of the country, industrial and sector situation of the company and its financial figures for the fair value of the share and based on this analysis formulates forecasts for the future value of a company. On the other hand, the technical analysis is the examination of past price movements to predict future price movements. Technical analysis uses complex equations, graphs and indicators, calculated on the historical time series in order to detect signals for evaluating the trend of a financial market. The third category includes the traditional linear econometric models based on the method of regression such as AR, MA, ARIMA, ARCH, and GARCH. Most of them are trying to link a variable with its lag values or/and with other contemporaneous variables usually under the assumption of normality. The fourth category includes sophisticated nonlinear models from computational data analysis and machine learning statistical area such as neural networks, Bayesian networks, Support Vector Machines, random forests and hidden Markov models. It’s evident that applications researched in this study belong to the field of machine learning analysis of financial data, but have much in common with the principles of technical analysis and econometric times series analysis. Results in our experiments are evaluated in relation to methods that belong to categories 2 and 3.

The main principals behind the statistical framework and analysis embodied in the last three groups of methods are the following: the current price of an asset usually reflects all current and relevant information, the price fluctuations are not completely random, and that history repeats itself. All of the pre-mentioned principles are connected to the belief of inefficient functioning of financial markets. In other words, the philosophy of technical analysis, econometric models and pattern recognition models is based on the fact that various time series show some patterns in behavior and that often the hypothesis of random motion can be rejected i.e. the efficient market hypothesis does not always hold. Under the assumption of efficient markets stated by Eugene Fama [1; 2] trading systems based on the available information cannot produce higher yields than expected performance of an index in the long run. Furthermore according to the EMH the present value if the best predictor for the future value of an asset, and that prices follow a random walk. Consequently, recent econometric studies focus mainly on the analysis of volatility of financial time series instead of the price changes. This is because it is commonly accepted that the prediction of long-term financial market trends is difficult or impossible. This is based on the theory that the patterns usually disappear due to the forces of the free market. However this fact is not true when predictions are conducted for short periods of time. The shorter horizon does not permit direct adaptation to the new conditions leaving space for correct short-term predictions. Patterns also exist due to information asymmetry. It is evident, that information in financial market is not evenly distributed and this has implications for the different behavior of market participants. A well-informed investor is more likely to avoid the risks associated to an event from a less informed one. Although the theory of market efficiency supports that the market is anonymous and that all participants are equally well informed, this is rarely the case in the short term, granting the opportunity to some investors or investment firms to create models producing better returns. This is also supported by the fact that algorithmic trading has significantly expanded in recent years. In our thesis we focus mainly in the problem of daily forecasting in order to filter patterns that emerge from the inefficiency of the markets.

1.3 Why HMM - Motivation

The reasons leading to the selection of HMM as the main research topic for this thesis is qualitative and quantitative or theoretical, and are outlined below:

1.3.1 Flexible and enhanced modeling capabilities

Financial time series consist of multidimensional nonlinear data that make the process of modeling and accurate prediction difficult and burdensome. Specifically, there are three major difficulties for the accurate prediction of financial time series. First, patterns of known economic time series are dynamic continuously shifting in time, meaning there exist no single model constant throughout the duration of the data that captures efficiently all the information contained in the data. Secondly, it is difficult to distinguish the long-term trend and short-term movements in the modeled process. In other words, an effective system should be able to adjust its sensitivity through time, based on the increasing amount of the data analyzed. Third, it is usually difficult to determine the usefulness of information and to separate the noise from the essential information. HMM, combining data processing, machine learning and forecasting techniques, offer the necessary flexibility to address these complicated temporal properties and successfully simulate non homogeneous processes.

1.3.2 Physical interpretation - Information Asymmetry

It is a fact that all market participants in a financial system do not share the same information. Investment firms often have the ability to manipulate the market by having access to more information than a retail investor. Moreover, all market participants usually share different opinion for the current point in the economic cycle. Finally there are many factors that influence and shape the value of various assets such as psychology, supply and demand which are often not directly measurable and visible. The asymmetry of information described makes several forces affecting the evolution of financial markets invisible, i.e. hidden. The hidden forces that determine the stock price movement (like movements in large investment portfolios) or the unobserved underlying process that determines the exchange rate of two currencies leads naturally to the study of HMM for modeling financial time series.

1.3.3 Interpretation of the properties of asset prices

According to the research work in [3], Ryden analyzed the time series of daily returns of the index S & P 500 using a Gaussian Mixture HMM model and documented their ability to reproduce both the distributional properties of daily returns of stock values and the temporal properties of econometric time series. Specifically, distribution of log returns is completely different in various stages of the economy. For example, during recession data exhibit huge spikes with negative drift. These regimes switching behavior is captured by the underlying Markov process. Moreover, the long-run log returns follow a distribution with heavier tails compared to those of the normal distribution. A powerful solution is the use of mixture distributions with weights assigned to its various components. These mixture distributions families can produce a distribution that captures stylized features of data observed during economic downturns or market recovery. Their distribution gives various shapes reflecting levels of skewness and excess kurtosis. Consequently, in the current thesis mixtures of Gaussians or Student-t are employed in modeling.

1.3.4 The elegant and transparent mathematical theory

Fitted HMM, based on their theory, offer powerful statistical tools to analyze the time series:

1. Availability of all the moments: mean, variance, autocorrelations.

2. The likelihood function can be calculated with a relatively easy manner using specific algorithms and the required calculations order is linear with respect to the number of observations.

3. The marginal distributions are feasible and easy to identify and missing observations can be treated with small effort by using the properties of Markov chains.

4. The conditional distributions exist, and depend on the structure of the HMM

5. It is robust in modeling outliers using mixture distributions

6. Efficient algorithm exists for implementing a continuous online learning process.

1.4 Main Contribution of the thesis

In this thesis we thoroughly investigate HMMs and their relationship to models finance problems. The contribution of this research thesis is threefold. First we perform an extensive review of HMM theory and applications. Moreover, it aims to summarize the most significant applications of HMM in general and with special focus in the field of finance. We offer a thorough and compact summary of the uses and the results of HMM in the last decades.

Secondly, we extend the framework of HMMs by proposing a theoretical variation which offers greater flexibility in their structure. Hidden Markov models (HMMs) are popular approach for modeling sequential data, typically based on the assumption of a first- or moderate-order Markov chain. However, in many real-world scenarios the modeled data entail temporal dynamics the patterns of which change over time. In this thesis, we address this problem by proposing a novel HMM formulation, treating temporal dependencies as latent variables over which inference is performed. Specifically, we introduce a hierarchical graphical model comprising two hidden layers: on the first layer, we postulate a chain of latent observation-emitting states, the temporal dependencies between which may change over time; on the second layer, we postulate a latent first-order Markov chain modeling the evolution of temporal dynamics (dependence jumps) pertaining to the first-layer latent process. As a result of this construction, our method allows for effectively modeling non-homogeneous observed data, where the patterns of the entailed temporal dynamics may change over time. We devise efficient training and inference algorithms for our model, following the expectation-maximization paradigm. We demonstrate the efficacy and usefulness of our approach considering several real-world datasets. As we show, our model allows for increased modeling and predictive performance compared to the state-of-the-art in the considered scenarios, for competitive computational complexity.

Finally we investigate the HMM efficiency in the problem of corporate credit scoring. Corporate credit scoring is widely used by financial institutions for portfolio risk management, and for pricing financial products designed for corporations. In addition, from a regulatory perspective, internal rating models are commonly used for establishing a more risk-sensitive capital adequacy framework for financial institutions. In this context, a large variety of statistical and machine learning tools have been applied to allow for successfully distinguishing between good and bad obligors. In this work, we propose a novel corporate credit rating system based on Student's-t hidden Markov models (SHMMs). SHMMs are a well established method for modeling heavy-tailed time-series data. Under our approach, we use a properly selected set of financial ratios to perform credit scoring. For each one of these financial ratios, we postulate a distinct SHHM, trained on five-year time-series data. Eventually, we aggregate the prediction signals generated by these SHMMs, using a linear predictive model optimized by application of an efficient genetic algorithm. We evaluate our method using a dataset pertaining to Greek corporations and SMEs; this dataset includes five-year financial data, and delinquency behavioral information. We perform extensive comparisons of the credit risk assessments obtained from our method with other broadly-used models, namely methods based on feed-forward neural networks, random forests, support vector machines, linear discriminant analysis, logistic regression, and Chi-squared Automatic Interaction Detector (CHAID). As we show, our approach yields better and more stable discriminatory performance in the considered scenarios compared to the considered state-of-the-art alternatives.

1.5 Structure of the thesis

The rest of the thesis is organized as follows: In the second chapter the fundamentals of HMM's theory are presented including main implementation issues, forecasting under an HMM framework and evaluation methods for their efficiency. In the third chapter an extended review of known HMM application are described with main concentration in the field of finance. In the same chapter a list of known variations of HMM structures are outlined. Chapter 4 presents in detail the novel theoretical HMM structure proposed in this thesis, called VDJ-HMM. The estimation and inference algorithms are thoroughly described and the relevant experimental results are outlined. Chapter 5 deals with the development of a novel credit rating system using Student-t Mixtures HMM as main component for its set up and outlines the experimental results of its efficiency against other state of the art models like Neural Networks and logistic regression. A summary of findings and conclusions are considered and possible extensions of the research work of this thesis are given in chapter 7. Furthermore other areas worth exploring referring to the application of HMM in finance are also described in the last chapter.

Chapter 2 Hidden Markov Models Fundamentals

2.1 Hidden Markov Models

A Hidden Markov Model is a statistical model in which the financial system under investigation is assumed to follow a non-visible Markov chain. Furthermore it is assumed that this variable affects another variable or variables that their price is observed and measured. Thus, an HMM's structure offers a flexible and general purpose model framework for univariate and multivariate analysis, mainly for discrete time series and classification data. Moreover, HMM are considered as a special group class of models for mixture distributions.

In a regular Markov chain the value of the variable modeled is visible and therefore the only thing that one needs to assess is the transition probabilities between the possible states of the system. In a Hidden Markov Model the state is not directly visible and we infer its changes only through the observable variables that they affect. More specifically for each state of the hidden variable a probability distribution is allocated for the observed variable and based on the sequence in the latter we can estimate theoretically the most likely sequence that created it. The term hidden refers to our inability to observe the actual state of the system, even if we know the probability transition matrix of the Markov process. This chapter provides a brief introduction to HMMs and a description of the basic theory around these models.

The diagram below shows the general architecture of a simple HMM. As shown the value of the variable which can be observed O depends on the value of the hidden variable Q.

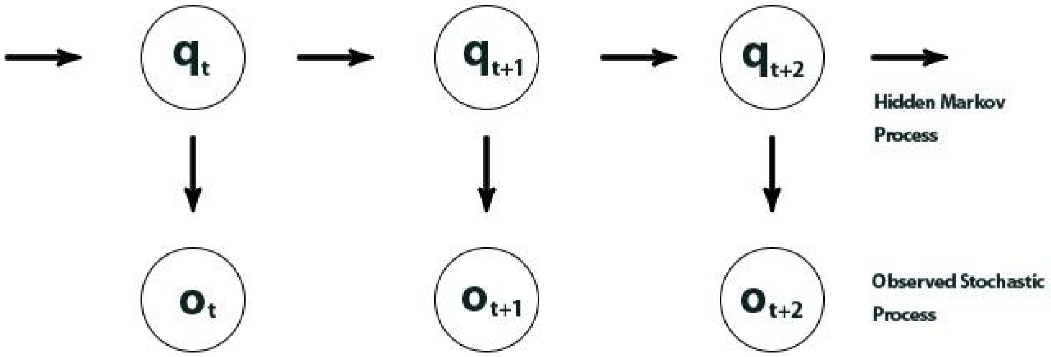


Figure 2.1: Hidden Markov Models graphical representation

By modeling a system and assuming that the data were generated by an HMM process, various algorithms exist to estimate the parameters of the system like the transition matrix of the hidden process, the conditional distributions of the visible variable based on the state of the hidden variable, and the marginal distribution of the initial state of the hidden variable. A learning process can be established for the model to adapt to new observations and improve predicting efficiency. A more detailed presentation of HMM is available at [4].

2.2 Definition

A hidden Markov model consists of two stochastic processes: one is measurable and observed while the second underlying stochastic process is not visible but hidden and the only way to be inferred is by observing the first sequence. These two stochastic processes have the following properties:

a. The hidden stochastic process follows a Markov chain that characterizes the state of the system at time t. Thus the hidden process satisfies the Markov property

$$P(Q_{t+1} = q_{t+1} | Q_t = q_t) = P(Q_{t+1} = q_{t+1} | Q_t = q_t, \ Q_{t-1} = q_{t-1}, \dots, Q_0 = q_0)$$

b. The observed stochastic process depends entirely on the state of the hidden stochastic process and satisfies the conditional independence property. This property is described by the following mathematical relationship:

$$\mathrm{P}(O_t = o_t | O_0^{t-1} = o_0^{t-1}, Q_0^t = q_0^t) = \mathrm{P}(O_t = o_t | Q_t = q_t)$$

where $O_0^{t-1} = (O_{t-1}, \dots, O_0)$

A discrete hidden Markov model is characterized by the following parameters:

1. The number of states (N): The number of possible hidden system states. Although the states of the system are not visible many times there is a physical interpretation to determine their number. Typically the states are interconnected so that from any state, the system can migrate to any other state thus the Markov process is ergodic .We symbolize {1, ..., N} the different states and q_t the hidden state at time t
2. The number of distinct values of the visible Stochastic Processes (M) for each state of the Markov chain. It is the set of possible values of the variable of the system we observe. We denote the number of possible values as V = $\{\mathrm{v}_1, \dots, \mathrm{v}_\mathrm{M}\}$
3. The matrix of transition probabilities: For the Markov chain that characterizes the hidden states we denote by A = $\{\alpha_{\mathrm{ij}}\}$ the transition matrix where

$$\alpha_{ij} = P[q_{t+1} = j \mid q_t = i],\ 1 \leq i,\ j \leq N.$$

For the special case where it is possible to transition from any situation in any other state in one time step then $\alpha_{ij} > 0$ for all i, j. For some specific types of HMM by assumption, α_{ij} can be set to 0 for one or more pairs (i, j).

4. The conditional on the hidden states distribution of the observed variable: If we denote by B = $\{b_j(k)\}$ the distribution then

$$b_j(k) \ = P[o_t = k \mid q_t = j],\ 1 \leq j \leq N \text{ and } 1 \leq k \leq M.$$

For example, in the case of discrete HMM, $\{b_j(k)\}$ is the distribution of the discrete variable that we observe in the hidden state k.

5. The initial distribution of hidden states $\pi = \{\pi_i\}$ where

$$\pi_i = \ P[q_0 = i], \text{ for every } 1 \leq i \leq N$$

Usually we denote λ = (A, B, π) the group of parameters of an HMM. Obviously an HMM is fully defined by λ.

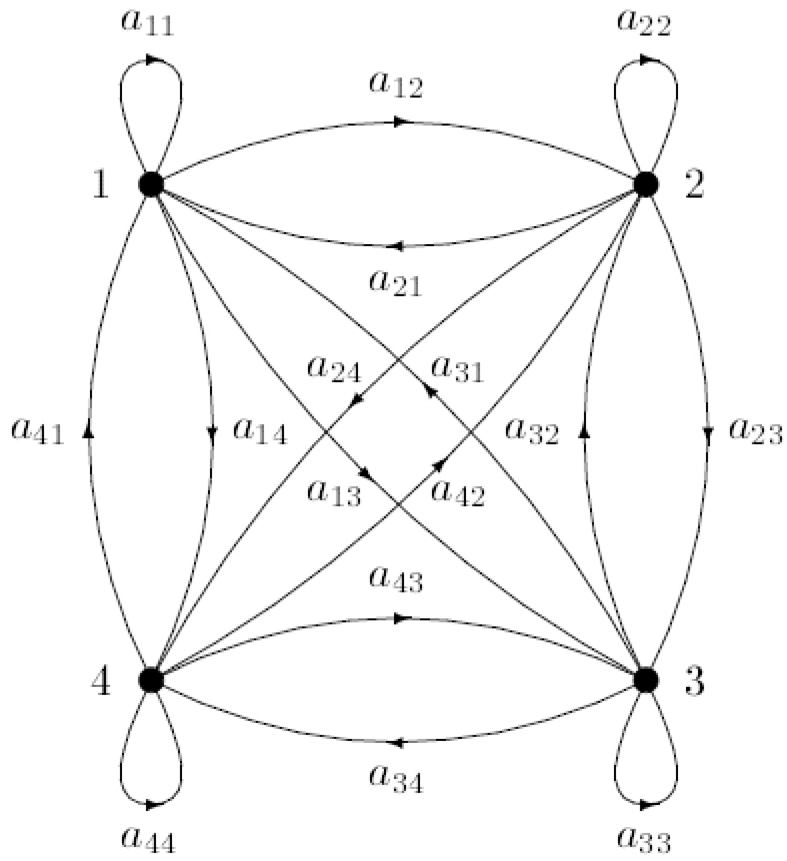


Figure 2.2: An ergodic Markov process of four states

In the above definition relationships governing the system refer to a discrete HMM that is an HMM that the conditional distribution of the observed stochastic process is discrete. However a very frequent HMM structure, applicable to continuous variables, are continuous HMM in which the distributions corresponding to each state belong to a family of continuous distributions like normal, student-t etc. Thus, a Gaussian mixture Hidden Markov Model is a continuous Markov model with the distributions of the observations being mixtures of normal distributions.

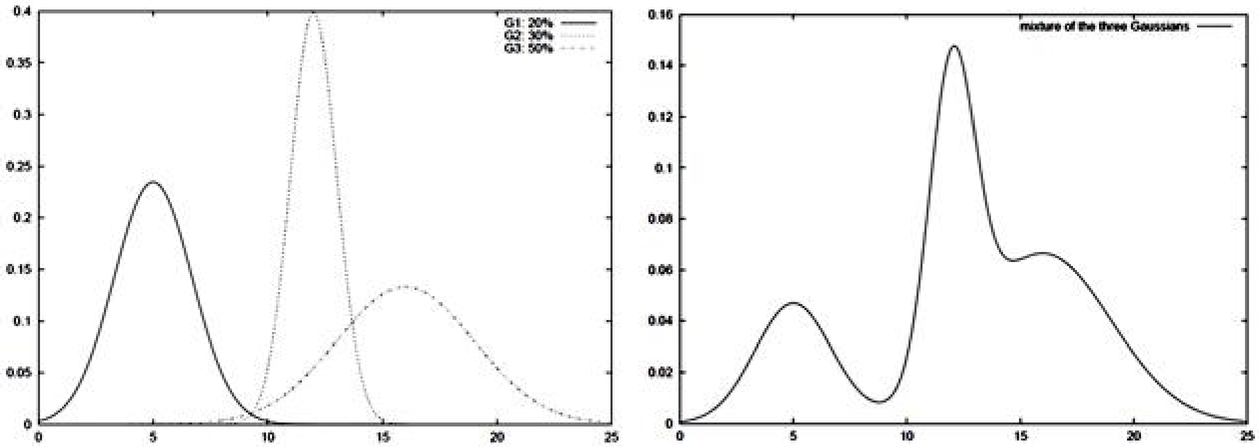


Figure 2.3: Normal Mixture Distribution with three components

Two options exist when applying an HMM in a data system where the observed variables are continuous: either use continuous distribution for the emission probabilities thus train a continuous HMM or apply a discretization process of the continuous observed variable mapping intervals to discrete values and subsequently estimate a discrete HMM. The procedure in the second case is obviously not taking into account all available information, however many times is preferable. For example, in the case where we are not interested in predicting the accurate level of the volume of transactions, but instead we want to predict whether in the next timestamp trading volume will be high or low we can discretize the volume variable. This process is called codebook.

2.3 Estimation problems of a hidden Markov model

Based on the definition of HMMs three basic problems emerge for their estimation:

The sequence likelihood evaluation: Given a sequence of observations O = $(o_1 , \ldots , o_T)$ and an HMM calculate the probability of this sequence occurring i.e P(O|λ) where λ = (π, A, B) is according to the definition.

The optimal state sequence inference: Given a sequence of observations O = $(o_1, \dots, o_T)$ and an HMM with parameters λ = (π, A, B) conduct inference on the optimum value S_t of sequence of the underlying hidden states that best interpret the sequence O for a given t ∈ 1,..., n or each n.

Parameter Estimation or Training Process: Given a sequence of observations O = $(o_1, \dots, o_T)$ and a specific structure of an HMM estimate the system parameters that maximize the likelihood function (maximum likelihood estimators) $\max_{\lambda^*} P(O|\lambda)$.

For these fundamental estimation problems of hidden Markov models specific algorithms have been developed to address them. These algorithms are described in detail below:

2.3.1 Sequence Likelihood Evaluation

The estimation of the probability of occurrence of the sequence O = $(o_1, \dots, o_T)$ of the observed variable can be considered as a scoring process signaling how well the estimated HMM interprets the specific sequence. In addition, using the likelihood of occurrence of a sequence, two HMM can be compared and evaluated based on their fitting efficiency to the data analyzed. To calculate the likelihood the sequence O = $(o_1, \dots, o_T)$ from the theorem of total probability that P(O|λ) = $\sum_{q \in Q} P(q|\lambda) P(O|\lambda, q)$ where q =$(q_0, \dots, q_T)$ is a likely sequence of hidden states and Q the set of all possible sequences corresponding to the hidden Markov chain. In the case where the different possible states q_i are N then the possible different sequences belonging in space Q are N^T. Based on the definition of an HMM $P(O|\lambda,q)=\prod_{t=1}^{T} b_{q_t}(o_t)$ holds, where $b_{q_t}(.)$ is the probability mass function or probability density function for the case of discrete or continuous hidden Markov model respectively. Moreover by definition of the Markov chain it holds $P(q|\lambda)= \pi_{q_0} \prod_{t=1}^{T} \alpha_{q_{t-1} q_t}$. Consequently, the following equation holds

$$P(O|\lambda) = \sum_{q \in Q} \pi_{q_0} \prod_{t=1}^{T} \alpha_{q_{t-1} q_t} \prod_{t=1}^{T} b_{q_t}(o_t) \quad (3.1)$$

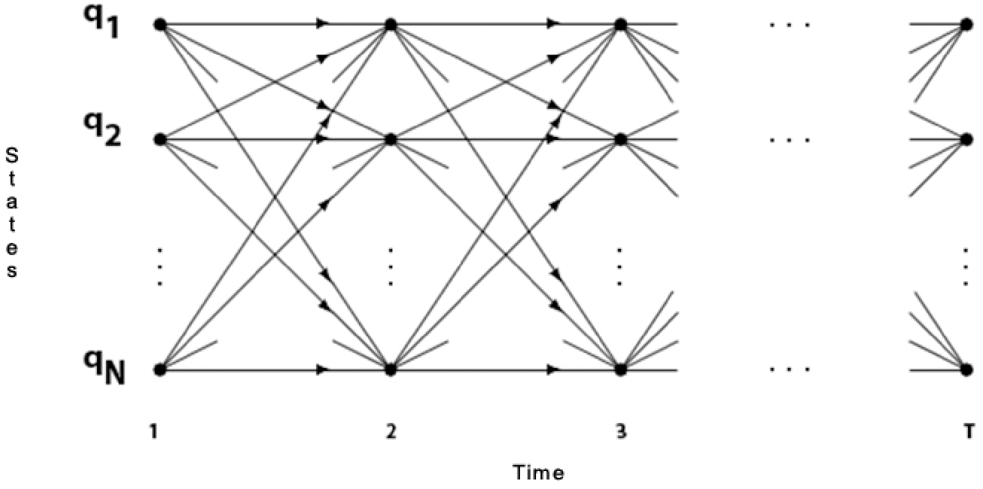


Figure 2.4: Graph of all possible transitions of the hidden states of an HMM

For the estimation of P(O|λ) by the definition of the relationship 3.1 the calculations required are of order (T * N^T), which means that even a moderate size HMM is highly computational intensive. Therefore to calculate the likelihood a more efficient algorithm is necessary. Such algorithm exists and is known as the forward - backward algorithm. Essentially consists of two separate algorithms the Forward and Backward steps where both calculate the relation (3.1) using order of computations (N^2*T), making them much more efficient and faster than the direct calculation of relationship 3.1. These algorithms first appeared in article [5].

The Forward algorithm is defined as follows:

First we define the variable $\alpha_t(i) = P[o_1, o_2, \dots, o_t, q_t = i \mid \lambda]$, i.e. $\alpha_t(i)$ denotes the probability of the sequence of observations $o_1, o_2, \dots, o_t$ (until the time t) and the state of the Markov chain at time t equal to i, given the parameter λ of HMM. The forward algorithm is summarized in the following steps:

1. Initialization of variables

$$\boldsymbol{\alpha_1(i)} = \pi_i * b_i(o_1) \text{ where i, } 1 \leq i \leq N$$

2. induction

$$\boldsymbol{\alpha}_{t+1}(\boldsymbol{j}) = \left[\sum_{i=1}^{N} \boldsymbol{\alpha}_t(\boldsymbol{i}) \alpha_{ij}\right] b_j(o_{t+1}) \text{ where j, } 1 \leq i \leq N \text{ and } 1 \leq t \leq T - 1$$

3. completion

$$P(O|\lambda) = \sum_{i=1}^{N} \boldsymbol{\alpha}_T(\boldsymbol{i})$$

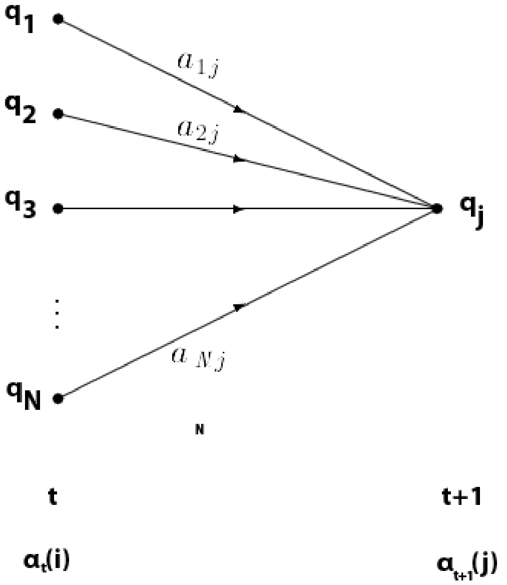


Figure 2.5: Graph of migration the forward algorithm

The first step sets the initial values for the forward probabilities as the joint probability of each state for time point 1 and the chances of observing o_1 given the individual hidden state. The second step (the step of induction) which is the most important step of the algorithm is shown in figure 2.6. The graph illustrates how the transition to state j at time t + 1 can be made from N possible states i, 1≤ i≤N, at time t. As $\alpha_t(i)$ is the probability for the occurrence of the data sequence $\mathrm{o}_1, \mathrm{o}_2, \dots, \mathrm{o}_t$ and condition that the hidden state at time t is i, then the product $\alpha_t(i)\alpha_{ij}$expresses the probability of the sequence $\mathrm{o}_1, \mathrm{o}_2, \dots, \mathrm{o}_t$; and the transition to state j at time t + 1 from state i at time t. By summing the product of all possible states N, i, 1≤ i≤N, at time t we derive the probability that the state is j at time t + 1 and the occurrence of the sequence $\mathrm{o}_1, \mathrm{o}_2, \dots, \mathrm{o}_t$

The Backward algorithm is defined in a similar manner as follows:

$$\boldsymbol{\beta}_{\mathrm{t}}(\boldsymbol{i}) = P[o_{t+1}, o_{t+2}, \dots, o_T | q_t = i, \lambda],$$

where $\beta_t(\mathrm{i})$ is the probability of occurrence of the partial sequence from time t + 1 until T given the state at time t is i and the parameters λ characterizing the trained HMM. In line with the forward algorithm for calculating $\beta_t(\mathrm{i})$ the following inductive steps are used:

1. Initialization of variables

$$\boldsymbol{\beta}_{\mathrm{T}}(\boldsymbol{i}) = 1, \; 1 \leq i \leq N$$

2. Induction

$$\boldsymbol{\beta}_{\mathrm{t}}(\boldsymbol{j}) = \sum_{j=1}^{N} \alpha_{ij} b_j(o_{t+1}) \boldsymbol{\beta}_{\mathrm{t+1}}(\boldsymbol{j}) \text{wherei, } 1 \leq i \leq N \text{ and } t = T-1, T-2, \dots, 1, \; 1 \leq i \leq N$$

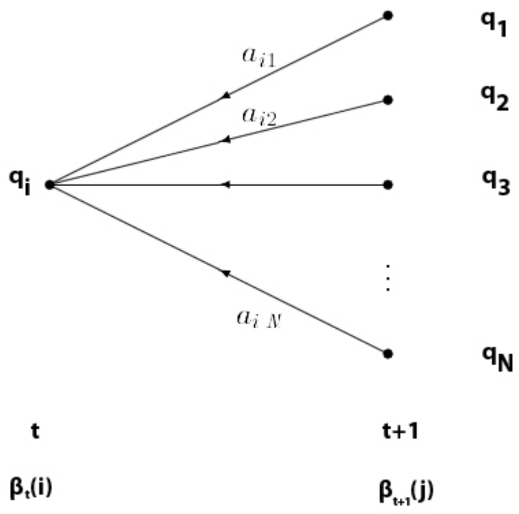


Figure 2.6: Graph transition of backward algorithm

The first step is setting arbitrarily all $\beta_T(i)$ equal to 1 for all i. In step 2, which is described in the figure 2.6, the system is in state i at time t and all possible transitions from state i to state j are depicted. Thus $\beta_t(j)$ is estimated based on the value of $\beta_{t+1}(j)$ estimated in the previous step and the probability of occurrence of the observation o_{t+1} for all possible transitions from state i to state j (α_{ij} from the transition matrix). The total number of computational operations for backward algorithm is of order N^2* T.

2.3.2 Inference of emitting state sequences

To inference the hidden state sequence we must first define what criterion determines the optimal sequence of hidden states that best interprets the sequence of observed variables. Assuming that the optimal sequence is the one that maximizes for every time point the likelihood of $o_1, o_2, \dots, o_T$. . this criterion maximizes the expected number of correct hidden states. For the application of this criterion we define the probability

$$\boldsymbol{\gamma}_{\mathrm{t}}(\boldsymbol{i}) = P[q_t = i \,|O, \lambda]$$

That γ_t(i) the likelihood that the hidden state at time t equal to i given the sequence of observations O and the parameters of HMM λ. Given the definitions of α_t(i) , β_t(i)

$$\boldsymbol{\gamma}_{\mathrm{t}}(\boldsymbol{i}) = P[q_t = i \,|O, \lambda]=$$

$$\frac{P[q_t=i, O \,|\lambda]}{P[O\,|\lambda]} =$$

$$\frac{P[q_t = i, O \,|\lambda]}{\sum_{i=1}^{N} P[q_t = i, O \,|\lambda]}$$

Because, $P[q_t = i \,|O, \lambda]$ is equal to $\alpha_t(i)\beta_t(i)$ we can write that

$$\boldsymbol{\gamma}_{\mathrm{t}}(\boldsymbol{i})= \frac{\boldsymbol{\alpha}_t(\boldsymbol{i})\boldsymbol{\beta}_{\mathrm{t}}(\boldsymbol{i})}{\sum_{i=1}^{N} \boldsymbol{\alpha}_t(\boldsymbol{i})\boldsymbol{\beta}_{\mathrm{t}}(\boldsymbol{i})}$$

Using the definition for γ_t(i) we derive that

$$q_t^* = \underset{1 \leq\ i \leq N}{\arg\max}[\boldsymbol{\gamma}_{\mathrm{t}}(\boldsymbol{i})], \quad 1 \leq \mathrm{t} \leq T.$$

Although the above definition maximizes the number of correct hidden state of the system (by selecting the most probable state for any time t), there are some problems with the final derived Markov chain.

For example, when the HMM transition matrix contain zero for some specific transitions ($\alpha_{ij} = 0$, for some i and j), then the optimal state switching sequence may not be valid if it contains one of these pairs. This is because this optimization criterion looks at q_t^* individually. Moreover, the sequence may not result in the maximum likelihood for the complete observed sequence O.

Finding the complete Markov chain sequence that maximizes the probability P(q | O, λ) which is equivalent to maximizing P(q, O |λ) leads to better and more acceptable estimates for the sequence of hidden states. A formal technique for finding the best path for the hidden states is based on methods of dynamic programming and is called the Viterbi algorithm.

For finding the best path $(q_1, \dots q_T)$ according to the algorithm given the sequence of visible variables $(o_1, o_2, \dots, o_T)$ first we must define the relationship:

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} \mathrm{P}\left[q_1, \dots, q_{t-1},\ q_t = i, o_1, o_2, \dots, o_t | \lambda\right]$$

$\delta_t(i)$ is the maximum probability derived for the sequence until time t, which corresponds to the first t observations and ends in state i. By induction occurs

$$\delta_{t+1}(j) = \max_{i} \left[\delta_t(i)\alpha_{ij}\right] b_j(o_{t+1}) \mathbf{(2)}$$

To calculate the optimal state sequence at each step the state that maximizes the equation (2) should be stored. For this reason we store this information in a table $\psi_t(j)$. The integrated process for finding the best path is summarized in the following steps:

1. Initialization of variables

$$\delta_1(i) = \pi_i b_\mathrm{i}(\mathrm{o}_1), 1\leq i\leq N$$

$$\psi_1(i) = 0$$

2. iteration

$$\delta_t(j) = \max_{1 \leq i \leq N}\left[\delta_{t-1}(i)\alpha_{ij}\right]b_j(o_t)$$

$$\psi_t(j) = arg\max_{1 \leq i \leq N}\left[\delta_{t-1}(i)\alpha_{ij}\right]$$

3. completion

$$P^* = \max_{1 \leq i \leq N}[\delta_T(i)]$$

$$q_t{}^* = arg\max_{1 \leq i \leq N}[\delta_T(i)]$$

4. The optimal sequence

$$q_t{}^* = \psi_{t+1}(q_{t+1}{}^*)\,, \qquad t=T-1, T-2, \ldots, 1.$$

Note that the Viterbi algorithm is similar to the forward algorithm with the exception of the optimization step and the step of the optimal sequence reversely estimated (step 4).

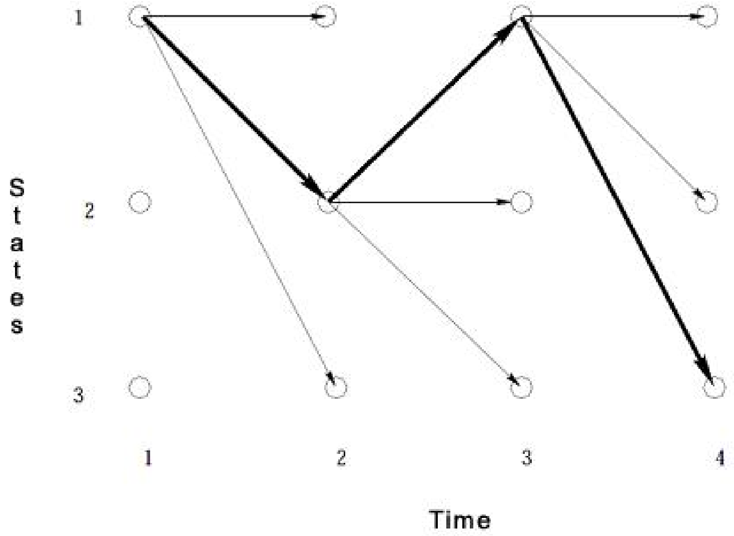


Figure 2.7: Graph of the optimal path based on the Viterbi algorithm

2.3.3 Training Process

The training process, relates to the definition of a method to estimate and update the parameters of an HMM, λ= (A, B, π) satisfying some specific optimization criterion. However, in the case of HMMs there is no analytical method for estimating parameter using closed form equations. The parameters are generally estimated using the method of maximum-likelihood (ML). The likelihood equations have a highly nonlinear structure and there is no analytical solution for the ML. The two most common approaches to estimate the parameters of an HMM are the EM algorithm and direct numerical maximization (DNM) of the likelihood. In this thesis we concentrate on the EM algorithm (Baum-Welch algorithm) for estimating the parameters of an HMM which usually yields better results. Below the Baum - Welch algorithm is summarized with details on the way it chooses the parameters of an HMM so as to maximize the likelihood function.

To describe the parameters of a training process of an HMM first we define the auxiliary variable $\boldsymbol{\xi}_{\mathrm{t}}(\boldsymbol{i},\boldsymbol{j})$ equal to the probability that the hidden state at time t is equal to i and at time t + 1 is equal to j, given the model λ and the observation sequence O:

$$\boldsymbol{\xi}_{\mathrm{t}}(\boldsymbol{i},\boldsymbol{j})= P[q_t = i,\ q_{t+1} = i|\ O, \lambda]$$

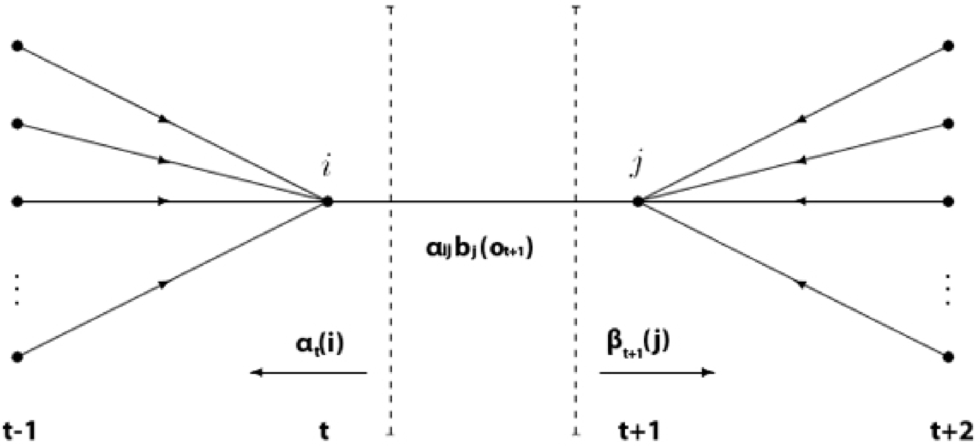


Figure 2.8: Graph of migration of the Baum - Welch algorithm

In figure 2.8 the basic transition step of the Baum-Welch algorithm is summarized. From the definitions of variables $\boldsymbol{\alpha}_t(\boldsymbol{i}), \boldsymbol{\beta}_{\mathrm{t}}(\boldsymbol{i})$,we derive

$$\boldsymbol{\xi}_{\mathrm{t}}(\boldsymbol{i},\boldsymbol{j}) = \frac{P[q_t = i,\ q_{t+1} = i, O|\lambda]}{P[O|\lambda]}$$

$$\boldsymbol{\xi}_{\mathrm{t}}(\boldsymbol{i},\boldsymbol{j}) = \frac{\boldsymbol{\alpha}_t(\boldsymbol{i})\alpha_{ij}b_j(o_{t+1})\boldsymbol{\beta}_{\mathrm{t+1}}(\boldsymbol{j})}{P[O|\lambda]}$$

$$\boldsymbol{\xi}_{\mathrm{t}}(\boldsymbol{i},\boldsymbol{j}) = \frac{\boldsymbol{\alpha}_t(\boldsymbol{i})\alpha_{ij}b_j(o_{t+1})\boldsymbol{\beta}_{\mathrm{t}+1}(\boldsymbol{j})}{\sum_{i=1}^{N}\sum_{j=1}^{N}\boldsymbol{\alpha}_t(\boldsymbol{i})\alpha_{ij}b_j(o_{t+1})\boldsymbol{\beta}_{\mathrm{t}+1}(\boldsymbol{j})} \text{ (2)}$$

The variable $\gamma_t(i)$ is the probability that the system is in hidden state i at time t given the model λ and the sequence of observations O, therefore based on the above definitions we get

$$\boldsymbol{\gamma}_{\mathrm{t}}(\boldsymbol{i})=\sum_{j=1}^{N}\boldsymbol{\xi}_{\mathrm{t}}(\boldsymbol{i},\boldsymbol{j}) = \frac{\boldsymbol{\alpha}_t(\boldsymbol{i})\boldsymbol{\beta}_{\mathrm{t}}(\boldsymbol{i})}{\sum_{i=1}^{N}\boldsymbol{\alpha}_t(\boldsymbol{i})\boldsymbol{\beta}_{\mathrm{t}}(\boldsymbol{i})} \text{ (1)}$$

Summing $\gamma_t(i)$ over t, we estimate the probability of the system being at state i. Similarly summing $\xi_t(i, j)$ for all times t we estimate the average number of times the system transitions from the state i to state j. therefore:

$\sum_{t=1}^{T-1}\gamma_t(i)$ = average expected number of transitions from state i corresponding to the sequence O

$\sum_{t=1}^{T-1}\xi_t(i, j)$= average expected number of transitions from state i to state j in the sequence O

Using the predefined variables we can define equations for the estimation of the parameters of an HMM.

$\pi_i^{'}$= probability that the system is in state i at time t = 1: $\boldsymbol{\gamma}_1(\boldsymbol{i})$ (3)

$a_{ij}^{'}$= (expected number of transitions from state i to state j) / (expected number of transitions from state i) = $\frac{\sum_{t=1}^{T-1}\boldsymbol{\xi}_{\mathrm{t}}(\boldsymbol{i},\boldsymbol{j})}{\sum_{t=1}^{T-1}\boldsymbol{\gamma}_{\mathrm{t}}(\boldsymbol{i})}$ (4)

$b_j(k)$= (expected number of transitions from state i and given that the observation is k) / (expected number of transitions from state i) = $\frac{\sum_{\mathrm{t}:\mathrm{o}_\mathrm{t}=\mathrm{v}_\mathrm{k},1\le\mathrm{t}\le\mathrm{T}}\boldsymbol{\gamma}_{\mathrm{t}}(\boldsymbol{i})}{\sum_{t=1}^{T}\boldsymbol{\gamma}_{\mathrm{t}}(\boldsymbol{i})}$ (5)

2.3.4 EM algorithm solving a discrete HMM

Summarizing all the above equations the final re-estimation algorithm is:

Counter of iterations: k: = 0

1. Set the initial values of the parameters of a discrete HMM
2. Calculation using the forward - backward algorithm of $\boldsymbol{\alpha_t(i)}$and $\beta_t(i)$ variables
3. E-step: calculation of $\boldsymbol{\gamma_t(i)}$, $\boldsymbol{\xi_t(i,j)}$ using the equations (1), (2), respectively
4. M-step: calculation of variables $\boldsymbol{\pi_j}$, $\boldsymbol{a_{ji}}$, $\boldsymbol{b_j(k)}$ using the relationship (3), (4), (5), respectively
5. If EM algorithm converges i.e. $\mathbf{l} - \mathbf{l}'$< threshold then stop else increase by 1 the iteration counter k: = k + 1 and repeat steps 2-5.

If we define an HMM model with parameters λ = (A, B, π) and use the equations defined above then we can transition into a new configuration for the HMM denoted by $\lambda' = (A', B', \pi')$. According to Baum Welch it holds that $P(O|\lambda') > P(O|\lambda)$ i.e. the new model is more efficient in interpreting the observation sequence O.

Note that the equations (3), (4), (5), are the same as the formulas obtained by applying the expectation maximization algorithm [6] with Q function

$$Q(\lambda', \lambda) = \sum_q P(O, q|\, \lambda')\log P(O, q|\lambda) \text{ δηλαδή,}$$

$$Q(\lambda, \lambda') = \sum_{q \in Q} \log \pi_{q_0} P(O, q|\lambda')$$

$$+\sum_{q \in Q}\left(\sum_{t=1}^{T} \log \alpha_{q_{t-1}q_t}\right) P(O, q|\lambda') + \sum_{q \in Q}\left(\sum_{t=1}^{T} \log b_{q_t}(o_t)\right) P(O, q|\lambda') \quad (2)$$

For a detailed documentation and derivation of the EM algorithm equations details are contained in references [7] [8].

2.3.5 Continuous HMM

Previously we described the theoretical approach to solve a discrete HMM, i.e. a HMM where the conditional distributions in hidden situations are discrete. However a very important group - class of HMM are the continuous HMM. The basic difference in the structure is that the conditional distributions of each state follow some continuous distributions.

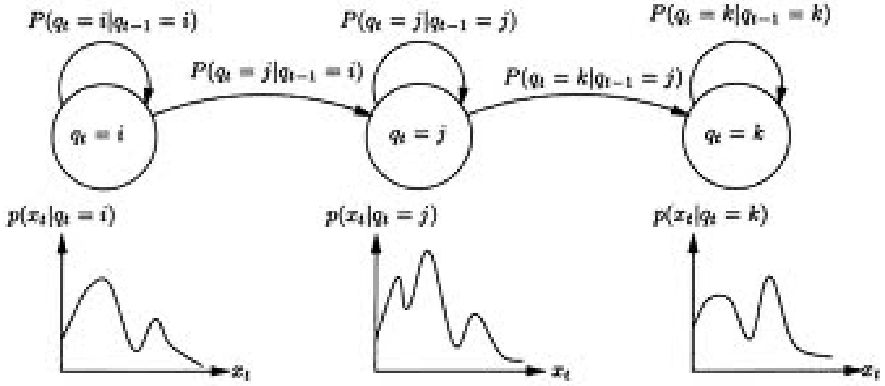


Figure 2.9: The evolution of an observed sequence following GHMM

Note that the table B in the continuous HMM parameters at λ = (A, B, π) correspond to the parameters of the continuous distributions assumed in the structure. Therefore equations solving the problem 3 on the transition matrix and the initial distribution of statements are the same as the discrete case. For the parameters of the distributions (i) equation (2) must be solved:

$$\theta' = \underset{\theta}{\arg\max}\left[\sum_{q\in Q}\left(\sum_{t=1}^{T}\log \mathrm{b}_{\mathrm{q}_\mathrm{t}}(\mathrm{o}_\mathrm{t})\right)\mathrm{P}(\mathrm{O},\mathrm{q}|\lambda')\right]$$

This is equivalent to the relationship

$$\theta' = \underset{\theta}{\arg\max}\left[\sum_{q\in Q}\left(\sum_{t=1}^{T}\log \mathrm{b}_{\mathrm{q}_\mathrm{t}}(\mathrm{o}_\mathrm{t})\right)\mathrm{P}(\mathrm{O},\mathrm{q}|\lambda')\right],\ i\leq t\leq T.$$

The most common structure for continuous distributions used in finance applications is mixture of normal distribution with a finite number of components. This is based on the fact that Gaussian Mixtures can reproduce all distributional properties of a financial time series based on Ryden [3]. In this case the form of conditional distributions is

$$b_j(o) = \sum_{m=1}^{M} c_{jm}\mathrm{N}\left(o_t, \mu_{jm}, \Sigma_{jm}\right), 1 \leq\ j \leq N$$

where $\sum_{\mathrm{j}=1}^{\mathrm{M}} \mathrm{c}_{\mathrm{jm}}$=1, o_t is the vector of observation we model and c_{jm} is the weight of m normal distribution component added to the mixture distribution corresponding to the state j and N is the known multivariate normal distribution.

$$\boldsymbol{b_{jm}(o_t) \sim \mathcal{N}\left(o_t, \mu_{jm}, \Sigma_{jm}\right) = \frac{1}{|\Sigma_{jm}|^{\frac{1}{2}}(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}(o_t - \mu_{jm})^T \Sigma_{jm}^{-1}(o_t - \mu_{jm})}}$$

Based on the literature the following equations define the auxiliary variables of the EM algorithm (Appendix 2) in the case of GHMM:

$$b_{jm}(o_t) \sim \mathcal{N}(o_t, \mu_{jm}, \Sigma_{jm}) = \frac{1}{|\Sigma_{jm}|^{\frac{1}{2}}(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}(o_t - \mu_{jm})^T \Sigma_{jm}^{-1}(o_t - \mu_{jm})}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\alpha_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{T}\sum_{j=1}^{N}\alpha_t(i)\alpha_{ij}b_j(o_{t+1})\beta_{t+1}(j)} \quad \textbf{(1)}$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{T}\alpha_t(i)\beta_t(i)} \quad \textbf{(2)}$$

$$\gamma_t(j,m) = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{T}\alpha_t(i)\beta_t(i)}\right]\left[\frac{w_{jm}\mathrm{N}\,(o_t,\mu_{jm},\Sigma_{jm})}{\sum_{k=1}^{M}w_{jk}\mathrm{N}\,(o_t,\mu_{jk},\Sigma_{jk})}\right] \quad \textbf{(3)}$$

$$w_{jm} = \frac{\sum_{t=1}^{T}\gamma_t(j,m)}{\sum_{t=1}^{T}\sum_{k=1}^{M}\gamma_t(j,k)} \quad \textbf{(4)}$$

$$\mu_{jm} = \frac{\sum_{t=1}^{T}\gamma_t(j,m)o_t}{\sum_{t=1}^{T}\gamma_t(j,m)} \quad \textbf{(5)}$$

$$\Sigma_{jm} = \frac{\sum_{t=1}^{T}\gamma_t(j,m)\left(o_t - \mu_{jm}^{new}\right)\left(o_t - \mu_{jm}^{new}\right)^T}{\sum_{t=1}^{T}\gamma_t(j,m)} \quad \textbf{(6)}$$

$$\pi_i' = \gamma_1(i) \quad \textbf{(7)}$$

$$a_{ij}' = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \quad \textbf{(8)}$$

EM Algorithm for Solving a GHMM

Using the above equations the EM algorithm for estimating a GHMM is:

Iterations Counter: k: = 0

1. Set the initial values of the parameters of an GHMM
2. Calculation using the forward - backward algorithm of $\boldsymbol{\alpha_t(i)}$ and $\beta_t(i)$ variables
3. E-step: calculation of the variables $\boldsymbol{\xi_t(i,j)}$, $\boldsymbol{\gamma_t(i)}$, $\boldsymbol{\gamma_t(j,m)}$ using the relations (1), (2), (3) respectively
4. M-step: calculation of variables $\boldsymbol{\Sigma_{jm}}$, $\boldsymbol{\mu_{jm}}$, $\boldsymbol{w_{jm}}$, $\boldsymbol{\pi_j}$, $\boldsymbol{a_{ji}}$ using the relations (6), (5), (4), (7), (8)

5. If EM algorithm converges i.e. $l - l' <$ threshold then finish else increase by 1 the iteration counter k: = k + 1 and repeat steps 2-5.

2.4 Issues related to the practical implementation of HMMs

Due to the complexity of HMMs, there are several practical issues related to the implementation of the algorithms forward - backward, Viterbi, and Baum - Welsh (Expectation Maximization) when estimating the parameters of a system. Below, the most important of these issues are summarized.

2.4.1 Thresholding or parameter values Limits

During the application of the algorithms forward – backward, the number of calculations can be reduced by setting a limit, where if the value of intermediate variables falls below this threshold they can be set to 0 without decreasing the efficiency of the model. Specifically, in the intermediate stages of forward, backward algorithms some $\alpha_t(i)$, $\beta_t(i)$ which are assigned very small values (less than a threshold C) it is practically proved that they can be set to zero without this affecting the performance of the HMM while significantly reducing the number of calculations due to the decrease in the terms in the intermediate sums estimated by the aforementioned algorithms.

2.4.2 HMM structure selection

When analyzing data using HMM, researchers should choose among numerous variations developed in academic literature, the appropriate structure for modeling the problem under investigation. However there is not a standard procedure to be followed in order to choose the ideal HMM structure in terms of the Markov chain properties, the number of hidden states, the type of conditional distributions, with covariates or not etc. Usually the choice depends on the already accumulated experience - literature in the respective field (finance, biology ...) along with any special features embedded in the sequences being analyzed. Especially in the case of GHMM one must additionally set the number of normal distributions components participating in each state conditioned mixture. The most frequent approach to address this issue is the development and training of various structures and selecting the best fit using some performance criteria like the Bayesian Information Criteria (BIC). Model selection in our experiments is extensively analyzed in chapter 4 and 5. It turns out that the optimal number of hidden states and mixtures components ranged between 2 to 6 under the GHMM set up from the experiments in this study.

2.4.3 Adequacy of data

In practice, it may be the case where the dataset available in a study is not enough to effectively estimate the parameters of an HMM. To address this issue in the case of multivariate Gaussian mixture HMM the following options are available: (i) assume parameter sharing across the Gaussians components, e.g. of the covariance matrices Σ, (ii) assume diagonal matrix for the covariance matrix Σ (γ) assume one Gaussian per hidden state. These options aim to decrease the number of parameters of the system in order to solve the issue of data limitations. In the discrete HMM case, when estimating the conditional distributions under a discrete HMM set up there is the chance that the combination q_t = i and d$o_t = v_k$does not exist in the data resulting in $b_j(k)$ = 0. This may affect the predictive ability of the model because of the zero estimates due to lack of sufficient numbers of data. In such cases the lack of data can be corrected by setting limits as minimum values of the parameters λ = (A, B, π). However when thresholds are applied the rest of the probabilities must be adjusted accordingly in order to sum up to 1.

2.4.4 Multiple data sequences

In many cases (e.g. left - right HMMs) HMM training is performed with the simultaneous use of multiple data sequences for the efficient estimation of parameters. Therefore the equations described previously in the training algorithms should be extended to take into account the parallel data processing of multiple observation sequences. This extension is relatively obvious and is described in detail [8]. Moreover, the way that each sequence affects the estimation of the parameters can be weighted to vary the significance corresponding to each time series of observations which may be due to different levels of reliability allocated to each sequence.

2.4.5 Scaling / Weighting

It is evident, that during the implementation of the training algorithms of HMMs intermediate variables $\alpha_t(i)$, $\beta_t(i)$ are decreasing converging to 0 with the increase in t. Consequently, due to the precision limitations of the computer used to implement the algorithms they are eventually set to 0. To avoid the rounding of variables to zero due to computational accuracy, weighting or scaling should be applied to each execution step of the forward - backward algorithms. The weighting process is described in detail in [8].

2.4.6 Initial values of variables

In theory, estimation algorithms lead to values of the parameters of HMMs that correspond to a local maximum of the likelihood function. Thus after selecting the structure of an HMM (number of states, type of conditional distributions, etc.), a key question is how to choose the initial parameters of the HMM so that the local maximum achieved through the implementation of the EM is equal or as close as possible to the total maximum of the likelihood function. From the literature it is evident that there is no simple and straightforward answer to this question. Rather, experience has shown that the random initial choice (subject to stochastic constraints on parameter values and the case for values different that zero) or use of uniform values for the parameters π, A, in almost all cases lead to good estimates of the systems under investigation. However, for the parameters B, studies have shown that good initial estimates (which have resulted from a specific method) is useful in the discrete case and play an essential role in the case of continuous distributions. These initial estimates can therefore be achieved in many ways such as for example the calculation of the empirical distribution followed by the observations after classified in subgroups (using algorithms from data categorization theory as k- means etc.) and estimate the mean and the variance of the observations within each class. In the context of this thesis will use a variant of the k-means algorithm specifically for HMM as described in appendix 4. Moreover, it should be stated that in the case of a continues learning process, with new time data, initial estimates may be derived from the best estimates of the parameters of the previous training step under the problem of financial time series analysis [9].

2.4.7 On-line learning Parameters

In forecasting financial time series new data gradually occur either periodically or ad hoc. This fact raises the question of the frequency of re-training of the parameters in the new data. The researcher has to decide between keeping the HMM up to date or establishing a learning process as computational efficient as possible. Obviously one of the criteria to take into account is the size of the sequence used for the training of HMM e.g. if data are stock prices per minute and the sequence length is 3 hours then the training process must be activated between 1 and 3 hours from the previous training step. Another criterion is the stability of the HMM's forecasting ability, i.e any significant deviation in the predictive accuracy that the system exhibited in the validation process (back testing) may trigger a retraining process. Furthermore techniques for exist in literature to address the restraining problem. On line learning of new data sequences allows adapting HMM parameters as new data becomes available, without having to retrain from the start on all accumulated training data. For example in [10] equations exist of the form

$$\lambda_n = f(\lambda_{n-1},\ o_n)$$

that offer the possibility of adjusting the parameters of the current parameterization $\lambda = (A, B, \pi)$ using the latest values of the observed sequence (o_n). With these estimates the renewal of the parameters is achieved based on the latest information while this process is less computational intensive than adjusting the parameters via the EM. A survey of techniques found in literature that are suitable for incremental learning of HMM parameters is outlined in [11].

2.4.8 Length of training observations / frequency time series data

For an analysis of daily observations the size of the training window resulting from empirical evidence usually ranges between 30-150 days length. However this is particularly true in univariate analysis while in the multivariate case the length window should be adjusted to account for the significant increase in the number of parameters of the system. In addition the final window size used for HMM training can be calculated by comparing the performance of various structure set ups with different lengths of sequences. On the other hand, the frequency of observations in the sequence analyzed, whether daily, hourly, weekly is mainly determined by the strategy or the forecasting problem we are addressing.

2.5 Forecasting using an HMM model

According to the literature, one of the main objectives of a data analysis with HMM is to develop forecast estimates of the modeled sequence. Specifically suppose we apply the structure of a GHMM in the return time series of a stock. In the case of the fitted HMM it is possible to predict the entire distribution of the variable analyzed and not only of a specific value. Therefore the information available to the analysts provide the opportunity of deciding between different risk-adjusted alternatives to follow. The following section describes possible approaches to conduct forecasting using HMM as they are studied and implemented in academic research:

2.5.1 Prediction based on the optimal path

First, using Viterbi algorithm the hidden state sequence that maximizes the likelihood function is estimated. Based on the states switching sequence derived by solving the decoding problem we can proceed to predict the distribution in the next time step in the following ways:

- Based on the state of the system at the last observation it is assumed that it will remain constant therefore this holds also for the distribution of the observations of the corresponding hidden state [9].
- Based on the optimal state corresponding to the most recent observation and considering the maximum probability in the transition matrix the most likely transition state is identified for the next time point. Then using the conditional distribution corresponding to this state the prediction of the next observation is performed [12]
- Based on the optimal hidden state corresponding to the most recent observation and considering all possible transitions, the weighted on the transition probabilities mixture distribution is

estimated. In the case of GHMM the forecast distribution is a mixture of mixture distributions with weights corresponding to all transition probabilities.

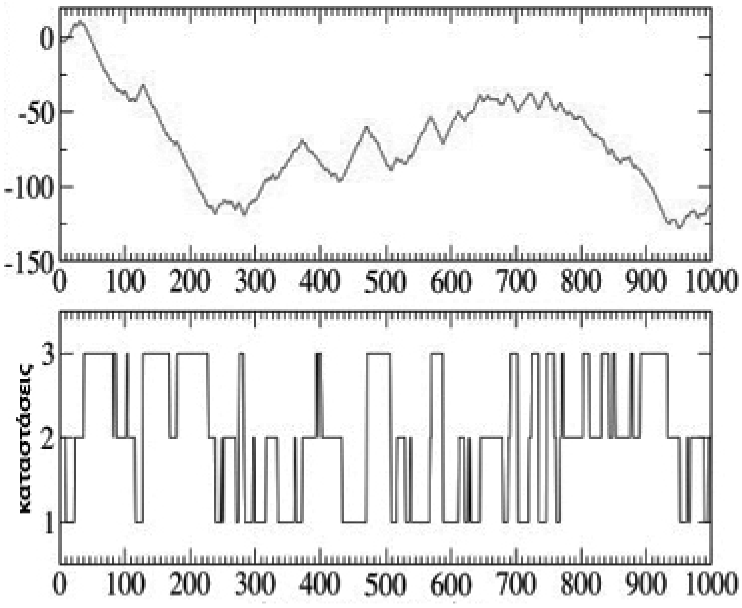


Figure 2.10: Decoding the switching of the hidden states for a stock forecasting problem

2.5.2 Prediction based on the forward algorithm

One way to conduct n-step forecasting using HMM is by utilizing the forward algorithm [13]. Specifically, using the forward algorithm the most probable hidden state, given the whole observation sequence, is estimated using the variables $\boldsymbol{\alpha}_{\mathrm{T}}(\boldsymbol{i})$. Based on this hidden state a migration distribution is estimated by combining the probabilities$\boldsymbol{\alpha}_{\mathrm{T}}(\boldsymbol{i})$, the transition matrix probabilities and the trained conditional distributions to conduct one step transition of the system. In experiments performed in this thesis in chapter 4 the forward algorithm was chosen as the most efficient way for forecasting.

2.5.3 Pattern identification approach

An alternative method for forecasting through HMM belongs to the area of pattern recognition. Specifically, after training in the most recent observation sequence an HMM using the forward algorithm we estimate the likelihood of historical sequences and filter the ones were their likelihood is within a narrow range with the current sequence' likelihood. After the identification of one or more historical points in the observation sequence making we aggregate the one step subsequent values of the variable and using an appropriate weighting we formulate a prediction for the future performance. This method with some variations can be found in [14; 15; 16; 17]. Finally, the search for pattern in the sequence of observations can be applied in the optimal states sequence using the Viterbi algorithm on historical data instead on the observable sequence.

2.5.4 Use two or more HMM for classification data

In the case of a binary prediction problem, for example the up/down movement of a stock then two HMMs are trained using multiple sequences were the price moved up or down respectively in the next time stamp. Subsequently for performing a prediction we score the current sequence through both trained hmm and based on the generated two likelihood values we infer the next stock movement. An example of this method is performed in [18]. Moreover this method of prediction is employed in chapter 5 for the development of an HMM credit classification system for companies.

2.6 Evaluation of Hidden Markov Models

From the literature it is evident that there is no standard evaluation framework for the HMM and their performance. Substantial differences exist in the case where the predictive power of an HMM, as part of a trading strategy, is assessed. Below we summarize a series of tests and indicators to be used in evaluating the performance of HMM.

2.6.1 Likelihood Ratio Test - Index Entropy

For comparison of two HMM with respect to better fitting the analyzed data sequence, the Likelihood Ratio Test can be used. The definition is given by the relationship:

Likelihood Ratio Test = (1) / T [log (likelihood of the HMM modeled of the null hypothesis) - log (likelihood for alternative model HMM)]

The Likelihood Ratio Test follows the X^2distribution with degrees of freedom the difference in degrees of freedom of the two HMM. This test is further known as Kullback-Leibler [8].

2.6.2 Penalized likelihood criteria BIC AIC

The Akaike Information Criterion (AIC) can be used for a comparison of two or more HMM. The index is given by the following equation AIC = -2 log L + 2p, and is an indication for best fitting of an HMM of the observation sequence. In this equation p represents the number of variables used in the HMM and L the likelihood of the data sequence. The Bayesian Information Criterion (BIC) can be used as a comparison of two or more HMM. The statistic is given by the following equation BIC = -2 log L + p logT, where T is the number of observations and p and L are the same as the AIC. [13]

2.6.3 Accuracy Prediction - Error Rate

For each implementation of an HMM we apply it to historical data sequences and calculate the percentage error in one step forward binary prediction. For this statistic confidence intervals can be produced using the mean and deviation estimated on historical sequences. The confidence intervals are compared in order to select the HMM which exhibits the lowest error rate. In chapter 4 in the experiments on return forecasting the error rate is used to determine the directional accuracy of the VDJ HMM.

2.6.4 Mean Square Error

Using the trained HMM it is feasible to predict the entire distribution of the variable in the next time step which enables us to calculate the mean squared error for a number of estimates and to compare it with other HMM set ups. Moreover this criterion can be used to evaluate how well the data are approximated by the trained HMM.

2.6.5 Annualized Return

If the HMM are applied in the framework of building an investment strategy their efficiency can be measured through the annualized return of the portfolio profitability. This statistic is calculated based on the average yield of the portfolio following the signals generated from an underlying HMM. Similarly, using this statistic estimated in various historical sequences the dispersion in the HMM's performance can be analyzed and the most profitable HMM candidate can be selected for supporting trading strategies. .

Chapter 3 Hidden Markov Models Applications and Variations

In the first part of this chapter various applications of HMMs are outlined with special focus in the financial sector and economic modeling. This chapter tries to cover the fields of applications of HMMs in order to serve as a reference point for future research endeavors. Listing application of HMM from other fields can also inspire new applications in finance especially when problem from different areas exhibit common properties and characteristics with financial time series. For each sector, we try to examine some of the most recent results and especially for the area of finance that is the subject of this thesis, the list of articles and results is more detailed. In the second part of this chapter a group of significant HMM variations, aiming to better capture the underlying properties of the frameworks they are applied to, are presented.

3.1 HMM Applications in various Fields

HMM were conceived and introduced for the first time in a series of statistical articles of Leonard E. Baum in the second half of 1960's. Specifically, HMMs appeared in the framework of a model for ecology [5] with their first name being «probabilistic functions of Markov chains». Since then they have been applied in numerous fields for data and signal analysis and pattern recognition. First, in 1970 they were implemented successfully in voice and speech recognition problems. Today, most of the commercial speech recognition systems are based on some variant of an HMM. In the 80s the implementation of HMMs was extended in the analysis of biological sequences such as DNA. Since then their importance increased significantly in the field of Bio-informatics. In the 90s the interest in the theory and applications of HMMs has spread in many fields due to their generalization abilities. Nowadays, Hidden Markov Models are also known for their application in temporal pattern recognition such as speech, handwriting, recognition gestures etc. In recent years HMM are also applied in fields like e-commerce for price forecasting of products sold via internet, to analyze the protein structure in molecular biology [19] and to predict oil prices [20]. Below is a short list of research areas of HMM applications and some illustrative studies applied.

3.1.1 Speech Recognition

As mentioned above, the initial researched majority of applications of HMM appear in speech recognition field. The literature on this subject is rich with thousands of conferences and publications in magazines and it would be almost impossible to outline their evolution up to date. Especially for this field the most important theory and results are included in the following articles [4] and [21] and in the following book [8]. According to the relevant theory, a separate hidden Markov model could be designed for each word in a vocabulary. Thus each HMM corresponds to different sounds like vowels, syllables and converts each audio signal to the respective words.

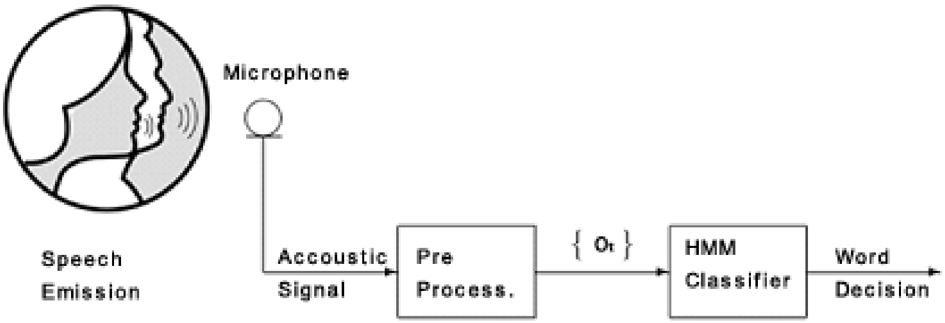


Figure 3.1: Recognition of isolated words using an HMM classifier

3.1.2 Computer Vision and Image processing

The HMM while originally developed in speech recognition systems in recent years have been implemented in many computer vision applications. Today successfully they are applied to text recognition systems, facial gestures, signatures and image processing. For example, in [22] HMM is applied in analysis of Car video real-time detection of front vehicles to support safer driving system. Furthermore in [23] a comparison of the performance of HMM for the recognition of handwritten Arabic words against dynamic Bayesian systems is performed. HMM have been applied as well in gesture recognition systems. An example of their applications in this area is described in [24]. Authors in [25] train an HMM for face recognition from video analysis. Although, images are by nature two-dimensional, and usually modeled by application of a Hidden Markov random field, it is possible to use the HMM for the processing of one-dimensional information of an image. For example in [26] a texture classifier based on the application of the HMM on image data converted by using wavelets is proposed. Furthermore, in [27] HMM is used to classify movements of people on a moving staircase into normal and abnormal. Finally, HMM have been applied for recognition of signatures [28] and for the handwritten text recognition combined with neural networks in [29].

3.1.3 Biomedical Applications

In the field of biomedicine, bioinformatics and biostatistics the HMMs have been applied in several sub fields such as DNA analysis, protein mixture and classifying a sequence of medical data such as electrocardiograms [30], visual stimuli [31], seizures [32]. In molecular biology, the hidden Markov models are used to model different evolution rates at different points on a molecular sequence[33]. A complete list of applications of HMM in the field of bioinformatics is contained in article [34] or in the book [35]. In [36] HMM were used for the analysis of protein sequences in different genes.

3.1.4 Epidemiology and Sociology

HMM have been applied to analyze the behavior of the sequence of animals under observation (kinetic behavior of locusts) [37]. In addition, the time series of homicides and suicides in Cape Town, South Africa, and birth data to a hospital in Africa have been analyzed with the use of hidden Markov models [38] and [13] respectively. In 2007 the hidden Markov models were applied to social conflict prediction in Indonesia [39].

3.1.5 Other Applications

In this section we list a number of applications that enhance the flexibility of HMM and spread in almost all scientific areas. In climatology, the appearance or not of rainfall in different locations can be modeled as an HMM where the climate conditions are not visible, leading to different distributions of rainfall per area and time period [40]. HMMs are used also by radar sonar for classification of underwater acoustic signals [26]. In 2005 HMM were applied to analyze the data of a sonar fitted to a robot modeling its possible movements [41]. In recent years the HMM are used to detect fraud in credit cards transaction by analyzing the behavior of payments [42]. Moreover, Hidden Markov models have been implemented both in telecommunications and in information technology with applications in cryptography and data decoding. A recent study analyzes using HMM security attacks in Skype [43] the most famous VoIP service online. In 2008 HMMs were applied to study the traffic on the internet in various protocols [44]. The classification of Human emotions into different people's profile was attempted in 2011 using HMM [45]. In 2009 HMM were used for classification of seismic active volcanoes analyzing their behavior [46].

3.2 HMM applications in finance

As we have already mentioned an HMM is a model that mixes different distributions based on an unobserved Markov process. This converts HMM as a strong statistical tool for time series modeling. Applications of HMM or Regime switching Markov models in finance are numerous and they first appeared in the late eighties. Today, HMM are used in finance with a wide range of applications like pricing derivatives, modeling of assets' volatility (stocks, indices ...), to estimate life insurance losses, forecast changes in interest rates and exchange rates etc. The most important applications of HMM in the field of finance are presented below in detail both historically and by sector.

3.2.1 Exchange Rates

The behavior of many exchange rate time series cannot be modeled efficiently with linear econometric models. Properties such as mean reversion, volatility of stock markets and the abrupt structural breaks are not efficiently captured using simple regression models. Thus, a range of applications using hidden Markov switching models have been researched in order to model the nonlinear properties of the exchange rates. These models belong to the family of HMM where according to their structure the parameters of the models are determined based on an underlying discrete Markov chains. Usually an AR, GARCH, ARMA models and their coefficients change according to the state of the system. In this vain Hamilton in 1989 presented an analysis of exchange rates against the dollar proving the existence of periods (regimes) of strengthening, and weakening of the dollar that last a long time [47]. Engel in 1994 [48] analyzed time series of exchange rates using HMM with mixed conclusions about their performance and forecasting efficiency. Specifically, the survey results showed that the predictive power for returns in currency exchange rates is low but there is evidence that the HMM can detect their long-term trend. In 2004, Cheung, Yin-Wong, Erlandsson, Ulf G., [49] analyze the time series of three exchange rates against the dollar and conclude that the efficient characterization of the different hidden states is significantly influenced by the frequency of the data and the length of the time series used to train the HMM. In [50] authors analyze time series of exchange rates and conclude that the Markov Switching Auto Regressive model best explains their non - linear relationships. Finally in 2008 [12] Patrik Idvall, Conny Jonsson applied a GHMM in the EUR / USD exchange rate for the purpose of building strategies for algorithmic trading, resulting in mixed results for their predictive ability.

3.2.2 Stock Market forecasting

An important area in finance is the analysis and prediction of the evolution of stocks and stock indices. In this area many research efforts to identify the interpretation of the forces that affect their movement using HMM exist. These applications are divided into two major categories: 1) analysis using only HMM for the prediction of time series and 2) analysis using a combination of statistical models and theories to improve the effectiveness of the HMM. In the first case a famous application of HMM is included in the published work of [3], who analyzed the time series of daily returns of the index S & P 500 using a GHMM model. This article documents the ability of GHMM to reproduce almost all features and stylized facts exhibited in the daily returns of stock indices and shares. In [51] the time series of six European stock indices are analyzed with HMM and the study concludes that mixtures of normal distributions best represent the times series of returns against the simple normal distribution while no clear differentiation in performance was evident with respect to the number of mixture’s components (2 or 3 normal distributions were investigated). In [52] and [53] the volatility of stocks using hidden Markov models is modeled and in [14] a new framework is presented for the implementation of HMM in predicting stock movements. Under this approach a multivariate GHMM is trained in the most recent sequences (open, close, high, low) of a share and then attempts to locate similar patterns in the historical data to predict the future behavior of stocks. The authors in their study make the assumption that the history of the performance of a stock is repeated. In 2010 two studies are presented, the first one concerns the valuation of European options [54] and the second one [55] who apply a hierarchical HMM (Hierarchical Hidden Markov Model) for predicting the Indian Nifty index with positive results. In the second group of studies in HMM literature hybrid models are presented where hidden Markov models are combined with some other statistical technique for the prediction of stock movements. Usually HMM are combined with neural networks and the theory of fuzzy logic. Specifically in [56] a model called 'Hidden Markov Experts' is applied according to which the projection is conducted through combining three Neural Networks (experts) and a Markov chain that is hidden and determines which of the 3 experts interpret the data better at each time point. Based on this structure, the parallel use of unsupervised and supervised learning is employed to estimate the parameters of the system. In 2007 HMM were applied in conjunction with neural networks for pattern recognition in the pricing of shares in [16]. Finally, in [15] Md. Rafiul Hassan presents a combination of HMM with a fuzzy logic model that shows promising results for predicting shares while its performance is superior in comparison with the statistical model investigated in [16]. In many cases the applications analyzed, showed positive signs for the ability of HMM to predict the rate of return for stock and stock indices, creating signals with good accuracy for investment purposes.

3.2.3 Other financial applications

HMM applications in finance were extended in the analysis of macroeconomic time series where Hamilton (1989) applied the methods of hidden Markov models for analyzing the rate of postwar GDP growth of the US using ARMA models for each hidden state in [57]. In the field of actuarial science in 2000 Poisson HMM where used to analyze the number of insurance claims in [58]. Moreover, in [59], hidden Markov models were used for the pricing of bonds by modeling the movement of the short interest rate. The shift in interest rates varied based on an underlying hidden Markov process. This way, the projections of short-term interest rate for the next period were estimated. In 2010 discrete HMM were used to forecast oil prices in [20]. Furthermore [60] using data from heating oil futures propose a model for the evolution of arbitrage-free futures prices under a HMM framework. Finally, applications of HMM have been implemented in the scientific area of credit risk mainly for portfolio modeling or simulating the actual credit quality of corporations. A thorough list of applications in credit risk is outlined in chapter 5 where a novel HMM rating system for credit rating is introduced within the context of this thesis.

3.3 HMM Variations

It should be noted that the idea, the nature and structure of an HMM differs significantly in various real life applications. In some cases HMM are well defined with natural meaning (i.e. hidden states correspond naturally to states of the system under consideration), while in other cases the definition of the hidden sequence is less clear, and there are cases where the hidden Markov chain meaning is fictional i.e. has no real meaning. In these cases, HMM are applied due to their increased mixture distribution capabilities for modeling time series data. In addition, depending on the problem different structures where developed and analyzed in order to capture effectively the properties of the underlying data. HMM variations are presented below along with a brief description:

3.3.1 Discrete - Continuous HMM

This category is extensively described in other parts of the thesis, but for completeness is included here with a brief description. The classification of an HMM as discrete [20] or continuous [9] depends on the type of the conditional distribution corresponding to the hidden states. However the term discrete – continuous HMM in the literature also refers to the time where the hidden Markov process is assumed to evolve, whether continuous or not. In this thesis we study mainly HMM having continuous distributions and a hidden Markov chain that changes in discrete time.

3.3.2 Ergodic - Stationary HMM

Ergodic - Stationary HMM are characterized depending on whether the Markov chain included in their structure is ergodic - stationary respectively. In this thesis, the HMM models applied, are assumed to comprise of an ergodic Markov chain to describe the evolution of hidden states.

3.3.3 Higher order or Weak HMM

In the case of Weak or higher order HMM the Markov chain describing the hidden process exhibit higher order than 1 i.e. the hidden state variable at time t + 1 depends on the hidden states not only at time t but also at t-1, t-2 etc. That is , the following relationship holds

$$P(Q_{t+1} = q_{t+1} | Q_t = q_t,\ Q_{t-1} = q_{t-1}, \dots) = P(Q_{t+1} = q_{t+1} | Q_t = q_t,\ Q_{t-1} = q_{t-1}, \dots, Q_0 = q_0)$$

In recent years, several studies indicated that financial time series exist that exhibit long memory which leads to the use of higher order Markov chains for HMM modeling [61].

3.3.4 Input - Output HMM or HMM with covariates

In this category the theoretical structure of HMM is extended to take into account external factors (covariates) affecting either the Markov process or the sequence of observed variables. Specifically, the HMM in this case can be interpreted as input - Output model (supervised learning) such as regression models or neural networks, where the observed variables are the dependent variables or the output variables. To convert HMM-based structure to an input model, exogenous data should be defined as input variables and the way affecting the design of a HMM must be defined. The external data can affect either the conditional distributions of the system or the chain transition matrix markov. Thus if we define a sequence of external data {Y_t} then the new parameters of HMM with covariates in the general form can be denoted as L = (p, A (Y), B (Y))

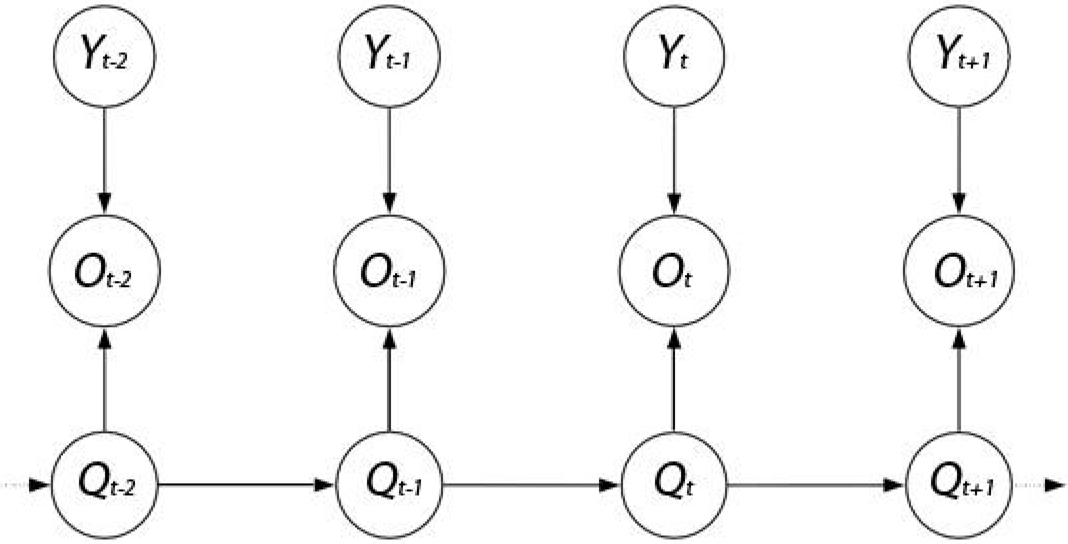


Figure 3.2: Graph HMM with covariates

Examples of HMM implementations with external covariates can be found in [62; 63]. Most techniques and algorithms for the estimation of the parameter for the classical structure of HMM can be easily extended to take into account external data. More about the structure of the input - output HMM and the theory about them can be found by Frasconi and Bengio, 1995 and 1996 in Articles [64; 65] respectively.

3.3.5 Left - Right HMM

The left right HMM assumes a specific structure in the Markov chain transition matrix of hidden states. Specifically, the transition matrix (A) is upper triangular. This determines the flow in states from left to right and assumes that the system remains in the same state or moves from this to the right and does not return again to this state. Additionally according to the transition matrix the Markov chain eventually leads to state N which is an absorbing state i.e. after reaching this state it cannot migrate to another state. More on Left-Right HMM can be found in [4]and their application in a financial series analysis is studied in [14].

3.3.6 Hidden Semi Markov Models

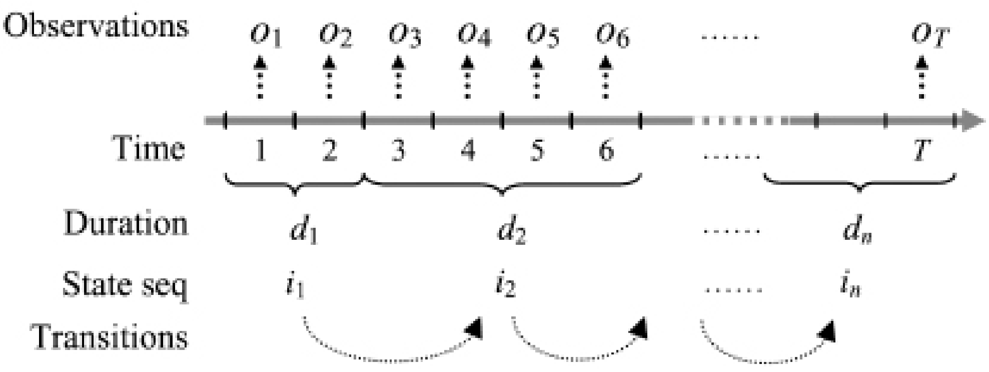


Figure 3.3: Graph of Semi HMM

The hidden semi-Markov model is an extension of the conventional hidden Markov model (HMM) which is useful for models where the underlying process stays in the same state for many consecutive time steps. In a conventional HMM the next hidden state at time t is determined only by the state at time t−1. This is usually represented as a state transition matrix A where each row represents a discrete distribution of the states that can be transitioned to from state i, hence the value a_{ij} represents this probability. This implies that the probability of remaining in the same state follows a geometric distribution. For models that stay in the same state for long periods of time the leading diagonal a_{ii} (representing self-transitions) will have the greatest probability mass. An alternative way to represent such a model is one where the state is represented by a state id i and a state duration d, both drawn from distribution a. This way the process describing the hidden states is considers semi Markov. For more information on the structure and the theory around the Hidden semi Markov models can be found in Articles [66; 67] .

3.3.7 Multistream fusion HMMs

A different probabilistic framework applied in multistream data is the multistream fusion HMMs approach [68], under which each stream is modeled separately using its own HMM. Then, analysis of the observed data can be conducted by creating a special HMM, recombining all the single stream HMM likelihoods at various specific temporal points. Obviously, depending on the specific selection of these recombination points, different solutions arise. For instance, in coupled hidden Markov models two component HMMs are linked by the dependence of their hidden states.

3.3.8 Markov Modulated Time Series or Regime Switching or Markov Regime Models or Autoregressive HMMs

This class of HMM is one of the most important variations particularly for their application in the analysis of financial series. Their structure differs from the classic definition of HMM in the fact that the observations (observed sequence) are modeled using an autocorrelation models like AR, ARCH, ARMA, ARIMA, GARCH and not a distribution. Essentially the basic assumption in such models is that the parameters in the regression of the time series change based on some hidden Markov process. That is the Markov chain is modeling the switch between regimes where each regime correspond to deferent coefficient of the assumed regression models. They are part of the broad concept of the family of HMM and they relax the conditional independence property of the observed sequence. However these models are considered as a special case of the continuous HMM because the dependent variable - observation follows a continuous distribution resulting from the autocorrelation equations and the assumed distribution of errors. These models were first introduced in finance in 1989 by Hamilton [57]. Since then variety of applications appear in economic publications about exchange rates [48; 49; 69] and stocks [51; 54; 70]. More detailed description of these model's properties and their estimation can be found in [71; 72].

3.3.9 Other HMM with more complicated structure

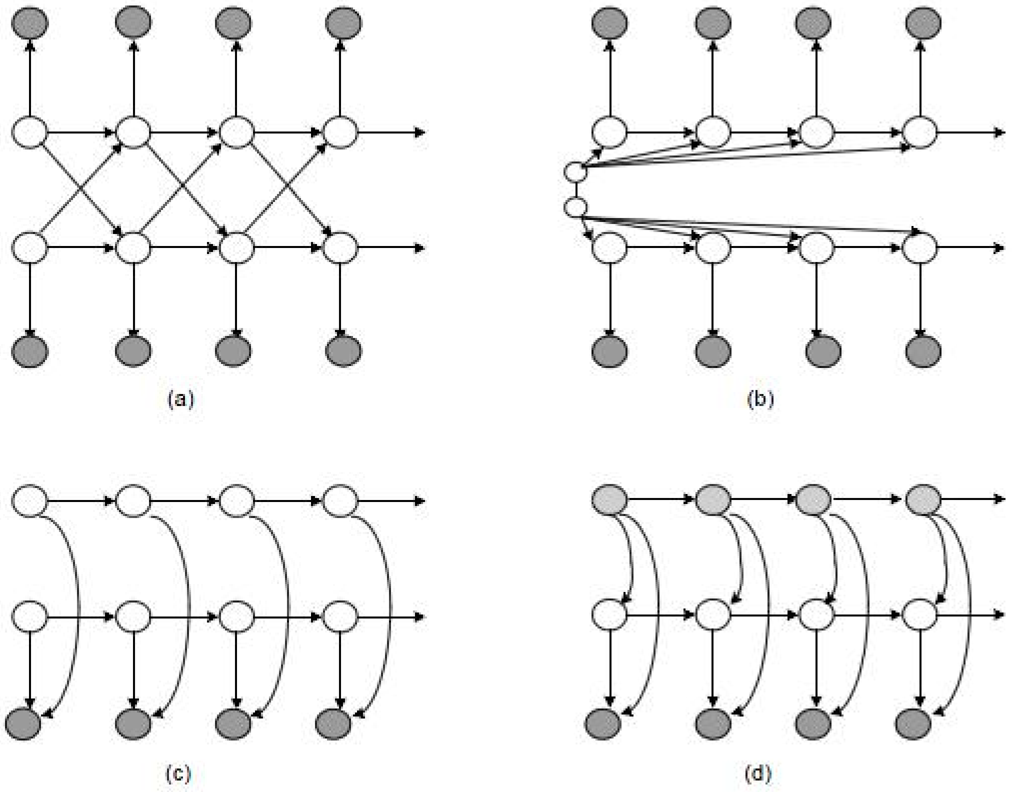


Figure 3.4: Different HMM structures. The white circles represent hidden states. The dark gray are observations processes and the light gray circles are input variables. a) Coupled HMM b) Event - Coupled HMM c) Factorial HMM d) Input - Output HMM

In the family of HMM a number of other structures - variations are belong which have been applied to complex problems, such as Hierarchical HMM [73; 55], Abstract Hidden Markov Models [74], Layered Hidden Markov Models [75; 76], Interactive Hidden Markov Models [77; 78] (Interactive between the hidden states and the observations relationship exist which transmits information both ways), Factorial HMM, Coupled HMM [79], Hybrid HMM (HMM + NN) and Hidden Markov Experts [16; 80; 81], Profile HMM [82] (with applications to the analysis of DNA and protein in bioinformatics)

Chapter 4 A variable order hidden Markov model with dependence jumps

4.1 Introduction

Modeling sequential data continues to be a fundamental task and a key challenge in the field of machine learning, encountered in a plethora of real-world applications, including bioinformatics, document analysis, financial engineering, speech processing, and computer vision, to name just a few. In this chapter, we focus on the problem of *sequence prediction*, dealing with *continuous*, possibly *high-dimensional* observations (time-series). Machine learning literature comprises a rather extensive corpus of proposed prediction algorithms for sequences of continuous observations. Among them, the hidden Markov model (HMM) is one of the most popular methods, used in a great variety of application contexts. This popularity is mainly due to the fact that HMMs are flexible enough to allow for modeling complex temporal patterns and structures in sequential data. Specifically, HMMs are popular for their provision of a convenient way of modeling observations appearing in a sequential manner and tending to cluster or to alternate between different possible components (subpopulations)[83].

Most popular HMM formulations are based on the postulation of first-order Markovian dependencies; in other words, only one-step-back temporal dynamics are considered. Such an assumption allows for increased simplicity and low computational complexity of the resulting model training and inference algorithms. However, postulating first-order temporal dynamics does also entail ignoring the possibility of the modeled data comprising longer temporal dynamics. Even though this assumption might be valid in some cases, it is well-known to be unrealistic in several application scenarios, including handwriting recognition, molecular biology, speech recognition, and volatility prediction in financial return series, thus undermining the modeling effectiveness.

To resolve this problem, several researchers have attempted to introduce HMM-type models with higher-order dependencies. Characteristic examples are the methods presented in [84]and [85], with successful applications to the problem of speech recognition, the method presented in [86], applied to handwriting recognition, the method of [87], designed to address challenges related to pattern recognition tasks in molecular biology, and the method presented in [88], which was successfully applied to the field of robotics. However, a major drawback of such higher-order HMM approaches is their considerably increased computational costs, which become rather prohibitive as model order increases. An effort to ameliorate these issues of higher-order HMMs is presented in [89]. In that work, instead of directly training R-th order HMMs on the data, a method of fast incremental training is used that progressively trains HMMs from first to R-th order.

Note, though, that using higher-order HMMs gives rise to a source of significant burden for researchers and practitioners, namely the need to determine the most appropriate order for the postulated models. This procedure entails fitting multiple models to the available data to choose from, and application of some cross-validation procedure, which, apart from computationally cumbersome, is also likely to become prone to overfitting [90]. Finally, another limitation of the existing higher-order HMM formulations concern their static and homogeneous assumptions, i.e. their consideration that the temporal dynamics order in the modeled data does not change over time. Indeed, sequential data with variable order in the entailed temporal dynamics are quite often encountered in real-world application scenarios [91; 92; 93; 94]. Therefore, allowing for capturing more complex structure of temporal dynamics in the modeled data, where effective model order may change over time as a result of dynamic switching between different temporal patterns, is expected to result in much better modeling and predictive performances.

To address these problems of conventional higher-order HMMs, some researchers have proposed appropriate models with variable order Markovian dynamics assumptions. For instance, a variable order Markov model is presented in [92] to address the problem of prediction of discrete sequences over a finite alphabet; the method is successfully applied to three different domains, namely English text, music pieces, and proteins (amino-acid sequences). More recently, [93] presented a simple and effective generalization of variable order Markov models to full online Bayesian estimation. Generalization of variable order Markov models in this context enables perpetual model improvement and enrichment of the learned temporal patterns by accumulation of observed data, without any need for human intervention. Despite these merits, a drawback of both these approaches concerns their inability to model sequential data comprising continuous observations, i.e. sequences each frame of which is a (probably high-dimensional) D-dimensional vector of real values, defined in $\mathbb{R}^D$. Finally, [95] propose a two-stage modeling approach towards variable order HMMs: the first stage consists in discovering repetitive temporal patterns of variable length, while the second stage consists in performing prediction by means of a separate simple HMM fit to the temporal pattern determined to be relevant at each specific time point. Similar to the previous approaches, a major limitation of [95]consists in its incapability to model sequential observations taking *continuous* values in $\mathbb{R}^D$.

In a different vein, a maximum-margin classifier for sequential data with (theoretically) infinitely-long temporal dependencies is presented in [91]. That paper devises a novel margin-maximizing model with convex objective function that allows for capturing arbitrarily-long temporal dependencies in sequential datasets. This is effected by utilizing a recently proposed nonparametric Bayesian model of label sequences with infinitely-long temporal dependencies, namely the *sequence memoizer* (SM) [96]. Training and inference for this model can be efficiently performed by employing a versatile mean-field-like approximation [97; 98], this approximation allows for increased computational efficiency, almost comparable to analogous (large-margin) first-order HMM formulations, e.g. [99]. Further, since design of the model of [91] is limited to classification tasks, a generalization of this model allowing to also perform sequence prediction was recently presented in [94].

As discussed in [91] and [94] the proposed models, postulating infinitely-long temporal dependencies, perform *inference* over temporal dependence patterns. In other words, they do *not* try to determine the most appropriate model setup for a considered dataset. Instead, they essentially learn a posterior distribution over all possible temporal dependence patterns. During prediction, these models effectively perform marginalization over all possible temporal dependence modeling assumptions, with each assumption being given a different probability at each time point. As a result, the methods in the aforementioned studies do not suffer from issues regarding appropriate model selection, namely the need to fit multiple models and perform cross-validation, and the associated overfitting proneness. In addition, they inherently allow for handling the case where the form of temporal dependencies changes over time. Note also that both the methods presented in [91] and [94] can model discrete as well as continuous observations, contrary to previous approaches which can handle only discrete observations.

Despite these merits, two major limitations of these approaches are:
(i) The need to come up with a (rather brute-force) approximation to allow for deriving efficient model training and inference algorithms, namely the mean-field approximation. Indeed, although mean-field approximation does not affect the nature of the model, which takes into account infinitely-long histories of latent temporal states, it results in omitting the fluctuations of higher-order temporal states when performing training and inference [91] and [94].This procedure leads to suboptimal training and inference results, that do not exploit the full modeling capacity of the methods, and with *no* theoretical convergence guarantees.

(ii) The need to perform inference for the employed model of (arbitrarily-long) temporal dynamics (state-transitions), i.e. a postulated sequence memoizer [100]. SM is a nonparametric Bayesian method recently proposed for modeling sequential data with discrete values and dependencies over infinitely-long time-windows. While effective in modeling sequential data with long temporal dynamics, inference for this model suffers from high computational costs, which increase with the length of the modeled sequences. As such, obviating the need of using an SM in the modeling pipeline is expected to significantly reduce computational costs.

In this chapter, we address all the aforementioned shortcomings of the current state-of-the-art, by introducing an HMM variant capable of capturing *jumps* in the temporal *dependence patterns* of modeled sequential data. Specifically, we introduce a hierarchical graphical model comprising two hidden layers: on the *first layer*, we postulate a *chain* of *latent observation-emitting states,* the *dependencies* between which may *change over time*; on the *second layer,* we postulate a *latent first-order Markov chain* modeling the *evolution* of temporal dynamics (*dependence jumps*) pertaining to the first-layer latent process. As a result of this construction, our model allows for effectively modeling non-homogeneous observed data, where the patterns of temporal dependencies may change over time. To allow for tractable training and inference procedures, our model considers *temporal dependencies* taking the form of *variable order dependence jumps*, the order of which is *inferred* from the data as part of the model inference procedure.

Our method is designed to allow for modeling *both* discrete and continuous observations; it allows for capturing seasonal effects in the modeled sequences, and enhances modeling in the implied autocorrelation structure of the observed sequences. In addition, contrary to the related methods of [91] and [94], our method does *not* require utilization of any kind of approximation to perform model training and inference. Indeed, both model training and inference can be performed *exactly* and in a computationally efficient way, using elegant algorithms derived under the expectation-maximization paradigm [6]. We demonstrate the efficacy of our approach in the task of sequential data prediction, considering real-world application scenarios.

The remainder of this chapter is organized as follows: In Section 2, we introduce our proposed model and derive its training and inference algorithms. In Section 3, we experimentally evaluate our approach, and exhibit its advantages over existing approaches. Finally, in Section 4 we conclude this chapter, summarizing and discussing our results.

4.2 Proposed Approach

4.2.1 Motivation

In real-world applications, it is often the case that stochastic processes are characterized by non-homogeneous evolution, exhibiting higher-order dependencies. For example, time series of financial asset returns are known to exhibit variable autocorrelation and non-stationarity[101], such forms of dynamics in the modeled data cannot be sufficiently captured by using a simple Markov process. In the same vein, historical volatility of financial asset returns usually exhibits long temporal interdependencies, slow autocorrelation decay, fat distribution tails, as well as temporal pattern switching over time, e.g. shifting between low volatility and high volatility regimes [102; 103; 104; 105] which are manifested as jumps driven by shocks or unexpected news [106; 107].

Several studies have examined whether conventional HMM formulations are capable of capturing such stylized facts in modeled time-series. For example, [3] examined the efficacy of simple first-order HMMs; further, [108] used hidden semi-Markov models (HSMMs) as an alternative solution allowing for better capturing the autocorrelation structure. However, the outcome of all these studies has been quite unsatisfactory compared to the state-of-the-art in the literature pertaining to the related applications, e.g. the literature on financial return series modeling. Motivated from these results, in this work we aim to come up with an elegant and computationally efficient HMM variant capable of accommodating the above-mentioned stylized facts in observed time-series, namely: (i) distributions with fat tails; (ii) seasonality and temporal clustering dynamics; and (iii) non-homogeneous temporal dynamics patterns, exhibiting dependence jumps over time.

4.2.2 Model Definition

As we have already discussed, in this work we are seeking to devise an HMM variant allowing for modeling sequential data with *variable temporal dependence patterns*, i.e. a model capable of determining *dependence jumps* in the chain of observation-emitting latent states. For this purpose, we postulate an HMM variant, the hierarchical construction of which comprises *two hidden layers*: The *first layer* essentially consists of the chain of *observation-emitting* latent states, the dependencies between which may *change form* over time. The *second layer* comprises a latent *first-order Markov chain* that determines (and generates) the *dependence jumps* taking place in the observation-emitting latent chain of the first layer.

Let us postulate N observation-emitting states on the chain of the first layer of our model, where the hidden emission density of each state is modeled by a M-component finite mixture model. Let us also postulate a latent first-order Markov chain comprising K states on the second layer; K is essentially the number of alternative temporal dependence patterns considered on the first layer of the model. Even though multiple alternative configurations could be considered for the form of the modeled temporal dependence patterns of the first-layer observation-emitting chain, in this work we limit ourselves to pair wise latent emitting state transitions between the *current* emitting state and *some previous state that occurred at a time point a number of steps back;* this number of steps back is determined from the latent values generated from the *second-layer dependence jumps-generating Markov chain* of our model.

Let us introduce here some useful notation. We denote as $O = \{\mathbf{o}_t\}_{t=1}^{T}$an observed data sequence, with $\mathbf{o}_t \in \mathbb{R}^D$. The latent (unobserved) data associated with this sequence comprise: (i) the corresponding *emitting state* sequence $Q = \{q_t\}_{t=1}^{T}$, where $q_t = 1, \ldots, N$ is the indicator of the state the tth observation is emitted from; (ii) the sequence of *temporal dependence form* indicators $Z = \{z_t\}_{t=1}^{T}$ that indicate the pairwise emitting states transition that is relevant ("active") at time t, where $z_t = 1, \ldots, K$; and (iii) the sequence of the corresponding *mixture component* indicators $L = \{l_t\}_{t=1}^{T}$, where $l_t = 1, \ldots, M$ indicates the mixture component density that generated the tth observation. A graphical illustration of the generative model and the latent interdependencies assumptions of our model is provided in Fig. 4.1.

The above-described VDJ-HMM model comprises the set of parameters $\Theta = \{\Phi, \Psi\}$, where Φ denotes the parameters set of the emission distributions of the model, and Ψ denotes the set of parameters of the postulated latent processes pertaining to the observed data dynamics (first-layer process) and the dependence jump dynamics (second-layer process). Specifically, since the second-layer process is a simple first-order Markov chain, it comprises the parameters

$$\hat{\varpi}_k \triangleq p(z_1 = k) \qquad \textbf{(1)}$$

that denote the (prior) probabilities of the initial state of this Markov chain, and the parameters

$$\hat{\pi}_{kk'} = p(z_t = k' | z_{t-1} = k) \ \forall t > 1 \qquad \textbf{(2)}$$

denoting the transition (prior) probabilities of this Markov chain. From the above model definition, we observe that, if the transition probability $\hat{\pi}_{11}$ in the above-defined transition matrix $\hat{\boldsymbol{\Pi}}\triangleq[\pi_{kk'}]_{k,k'}$ is close to one, then the observation-emitting process of our model (first model layer) almost reduces to a first-order Markov chain. In this chapter, for simplicity we set $\hat{\varpi}_k = 1/\mathrm{K}\ \forall k$ and $\hat{\pi}_{kk'} = 1/\mathrm{K}\ \forall k, k'$; in other words, we consider all dependence forms a priori of equal probability. These assumptions, although relatively limiting, allow for deriving tractable and computationally efficient model training and inference algorithms, as we show further on. Appendix 1 presents analytically the general case for the first model layer, outlines the relevant training and inference algorithms, along with all the derivation formulas for the estimation of the parameters.

In a similar fashion, the postulated first-layer process of our model comprises the parameters

$$\varpi_i \triangleq p(q_1 = i) \qquad \textbf{(3)}$$

denoting the (prior) probabilities of the initial observation-emitting state, with $\boldsymbol{\varpi}\triangleq[\varpi_i]_{i=1}{}^{N}$. In addition, turning to the variable-form temporal dynamics of this process, we also introduce the set of *dependence form-conditional* transition (prior) probability matrices $\{\boldsymbol{\Pi}^k\}_{k=1}{}^{K}$, with

$$\boldsymbol{\Pi}^k \triangleq [\pi_{ij}{}^{k}]_{i,j=1}{}^{N} \qquad \textbf{(4)}$$

where

$$\pi_{ij}{}^{k} \triangleq p(q_t = j | q_{t-1}, \ldots, q_{t-k} = i; z_t = k) = p(q_t = j | q_{t-k} = i; z_t = k) \qquad \textbf{(5)}$$

In other words, we consider different (pairwise) state transition probabilities, depending on the inferred dependence form k (number of steps back) generated from the postulated second-layer process.

Having defined the latent processes of our model, with effective parameters set $\Psi = \{\boldsymbol{\varpi}, \{\boldsymbol{\Pi}^k\}_{k=1}^{K}\}$, we can now proceed to the definition of the (conditional on the first-layer states) emission distributions of our model. For this purpose, and in order to allow for effective modeling of continuous-valued observations, we postulate M-component finite mixture models, as we have already discussed. Specifically, in our work, *to allow for modeling distributions with fat tails,* we consider two alternative selections: (i) multivariate Gaussian mixture models, yielding

$$p(\mathbf{o}_t|q_t = i) = \sum_{m=1}^{M} w_{im} N(\mathbf{o}_t|\boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}) \qquad \textbf{(6)}$$

where$N(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$; and (ii) multivariate Student's-t mixture models, yielding

$$p(\mathbf{o}_t|q_t = i) = \sum_{m=1}^{M} w_{im} S(\mathbf{o}_t|\boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}, v_{im}) \qquad \textbf{(7)}$$

where $S(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma}, v)$ is a multivariate Student's-t distribution with parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, and v degrees of freedom. On this basis, the parameters set Φ yields $\Phi = \{w_{im}, \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}\}_{i,m}$ or $\Phi = \{w_{im}, \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}, v_{im}\}_{i,m}$, respectively. As discussed in [109], HMM-type models with Student's-t mixture emission distributions allow for better modeling sequential data stemming from populations with long tails, which are quite common in real-world application scenarios.

This concludes the definition of our model. We dub our approach the variable dependence jump HMM (VDJ-HMM). From Eqs. (1)-(7), the joint distribution of VDJ-HMM yields:

$$p(O, Q, Z|\Theta) = \hat{\varpi}_{z_1}\varpi_{s_1}\prod_{t=1}^{T-1}\hat{\pi}_{z_t, z_{t+1}}\prod_{t>1}\pi_{s_{t-z_t}, s_t}^{z_t} \qquad (8)$$

$$\times \prod_{t=1}^{T} p(\mathbf{o}_t|q_t = i)$$

Note that, as observed from (8), a major advantage from the computational point of view of the proposed VDJ-HMM model compared to higher-order HMM formulations (e.g., [84; 88; 89]) is the much fewer number of parameters postulated from VDJ-HMM. As a result, VDJ-HMM is capable of capturing seasonal effects in the modeled data while allowing for significantly more efficient training and inference algorithms compared to existing alternatives. In addition, the lower number of trainable parameters reduces the tendency of the model to overfitting, as well as the associated requirements in training data availability to ensure effective model training.

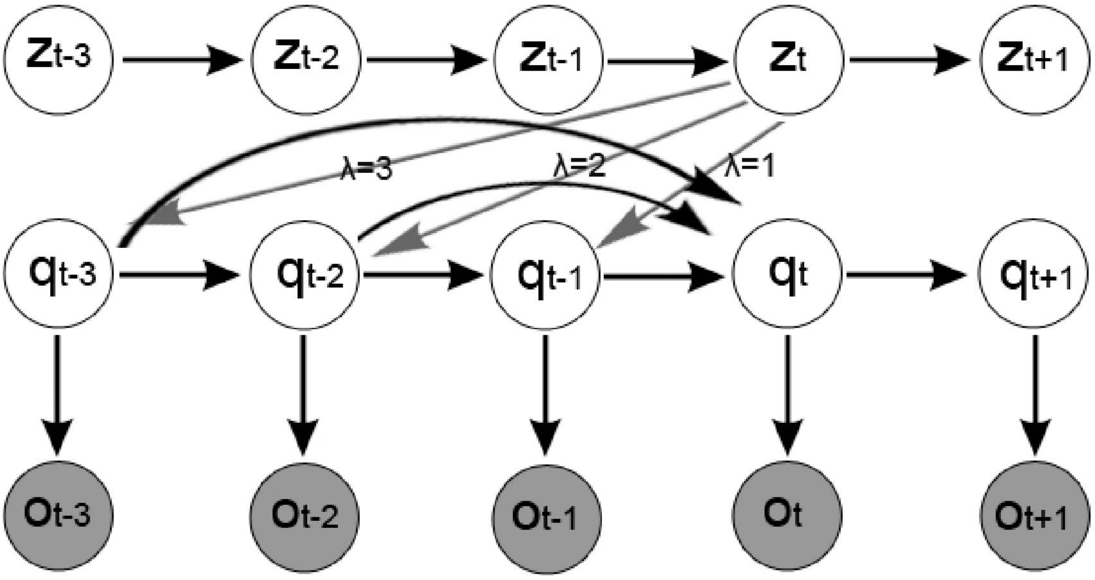


Figure 4.1: Graphical illustration of the generative model and the latent interdependencies assumptions of VDJ-HMM. Here, λ denotes the active dependence form inferred at time t by the model. Effectively, the value of λ determines which past observation-emitting latent state currently affects the temporal dynamics of observation generation.

4.2.3 Model Training

To perform training for our model given a sequence $O = \{\mathbf{o}_t\}_{t=1}^{T}$, we resort to the familiar expectation-maximization (EM) paradigm [generalization of the here-derived algorithm for the case of training with multiple sequences is straightforward]. Based on the definition of VDJ-HMM [Eqs. (1)-(7)], the complete data of our model comprise the observable sequence O, the corresponding emitting state sequence $Q = \{q_t\}_{t=1}^{T}$, the dependence form sequence $Z = \{z_t\}_{t=1}^{T}$, and the sequence of corresponding mixture component indicators $L = \{l_t\}_{t=1}^{T}$. In addition, based on the derivations of [109]in the special case of considering multivariate Student's-*t* mixture models as the emission distributions of VDJ-HMM, to allow for effective model training and inference procedures, we resort to expressing the multivariate Student's-t distributions as scale-mixtures of Gaussians, yielding [109]:

$$p(\mathbf{o}_t|q_t = i; \{u_{imt}\}_{m=1}^{M}) = \sum_{m=1}^{M} w_{im} \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}/u_{imt}) \qquad \mathbf{(9)}$$

where u_{imt} is a precision scalar corresponding to the observation $\mathbf{o}_t$ given it is generated from the jth component density of the ith emitting state, and is Gamma-distributed as [109]:

$$u_{imt} \sim \mathcal{G}\left(\frac{\nu_{im}}{2}, \frac{\nu_{im}}{2}\right) \qquad \mathbf{(10)}$$

Under this setup, the above introduced set of precision scalars $\{u_{imt}\}$ is also regarded as part of the complete data configuration of our model.

The EM algorithm comprises optimization of the posterior expectation of the complete data log-likelihood of the treated model [6]

$$Q(\Theta;\mathit{\Theta})\triangleq E_{\mathit{\Theta}}(\log L_c(\Theta)|O) \qquad \textbf{(11)}$$

where $\mathit{\Theta}$ˆ denotes the currently obtained estimator of the model parameters set Θ, and $\log L_c(\Theta)$ is the expression of the complete data log-likelihood of the model, which reads (ignoring constant terms)

$$\begin{aligned}\log L_c(\Theta) &= \textstyle\sum_{h=1}^{N}\Big\{\mathrm{I}[q_1 = h]\ \log\varpi_h \\ &+ \textstyle\sum_{k=1}^{K}\sum_t \mathrm{I}[z_t = k]\sum_{i=1}^{N}\mathrm{I}[q_{t-k} = h,\ q_t = i]\ \log\pi_{hi}^{k}\Big\} \\ &+ \textstyle\sum_{t=1}^{T}\sum_{i=1}^{N}\mathrm{I}[q_t = i]\log L_c(\mathbf{o}_t|q_t = i)\end{aligned} \qquad \textbf{(12)}$$

where I[·] is the boolean operator. In Eq. (12), $\log L_c(\mathbf{o}_t|q_t = i)$ is the complete data log-likelihood of the emission distribution of the ith hidden state with respect to $\mathbf{o}_t$, and the associated latent variables l_t and (in case of Student's-t models) $\{u_{imt}\}_m$. In the case of Gaussian mixture emission distributions, $\log L_c(\mathbf{o}_t|q_t = i)$ yields

$$\begin{aligned}\log L_c(\mathbf{o}_t|q_t = i) &= \textstyle\sum_{m=1}^{M}\mathrm{I}[l_t = m]\Big\{\log w_{im} - \frac{1}{2}\log|\mathbf{\Sigma}_{im}| \\ &. - \tfrac{1}{2}d(\mathbf{o}_t,\ \boldsymbol{\mu}_{im};\mathbf{\Sigma}_{im})\Big\}\end{aligned} \qquad \textbf{(13)}$$

where$d(\mathbf{o}_t,\ \boldsymbol{\mu}_{im};\mathbf{\Sigma}_{im})$ is the Mahalanobis distance between $\mathbf{o}_t$ and $\boldsymbol{\mu}_{im}$, with covariance matrix $\mathbf{\Sigma}_{im}$. On the other hand, in the case of Student's-t mixture emission distributions, $\log L_c(\mathbf{o}_t|q_t = i)$ yields

$$\begin{aligned}\log L_c(\mathbf{o}_t|q_t = i) &= \textstyle\sum_{m=1}^{M}\mathrm{I}[l_t = m]\Big\{-\log\Gamma(\frac{\nu_{im}}{2}) + \frac{\nu_{im}}{2}\times \\ &\Big[\log(\tfrac{\nu_{im}}{2}) + \log u_{imt} - u_{imt}\Big] + \log w_{im} \\ &. - \tfrac{u_{imt}}{2}d(\mathbf{o}_t,\ \boldsymbol{\mu}_{im};\mathbf{\Sigma}_{im}) - \tfrac{1}{2}\log|\mathbf{\Sigma}_{im}|\Big\}\end{aligned} \qquad \textbf{(14)}$$

where $\Gamma(\cdot)$ is the Gamma function.

As usual, the EM algorithm for our model is an iterative procedure, each iteration of which comprises an E-step and an M-step. On the E-step of the algorithm, we compute a set of posterior expectations pertaining to the latent variables of our model (sufficient statistics), using the current estimator of the model parameters set $\hat{\Theta}$. Subsequently, on the M-step of the algorithm, we optimize the model parameters set Θ using the sufficient statistics computed previously, in order to obtain an updated estimator of the model parameters set, $\hat{\Theta}$.

4.2.3.1 E-step

From (11) and (12), it directly follows that the E-step of our algorithm consists in computing the posterior probabilities of the latent states on the first and second hidden layers of our model, as well as the corresponding state transition posteriors. It also comprises computation of the emitting state-conditional mixture component posteriors, as well as the posteriors of the precision scalars u_{imt}, when considering Student's-*t* mixture emission distributions.

Let us begin with the mixture component posteriors, hereafter denoted as ξ_{imt}; we have

$$\xi_{imt} \triangleq E_{\Theta}(l_t = m | \mathbf{o}_t, q_t = i) = \frac{p(\mathbf{o}_t | q_t = i, l_t = m)}{\sum_{h=1}^{M} p(\mathbf{o}_t | q_t = i, l_t = h)} \qquad \mathbf{(15)}$$

This expression yields

$$\xi_{imt} = \frac{w_{im}\mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im})}{\sum_{h=1}^{M} w_{ih}\mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{ih}, \boldsymbol{\Sigma}_{ih})} \qquad \mathbf{(16)}$$

when considering Gaussian mixture emissions, and

$$\xi_{imt} = \frac{w_{im}\mathcal{S}(\mathbf{o}_t|\boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}, \nu_{im})}{\sum_{h=1}^{M} w_{ih}\mathcal{S}(\mathbf{o}_t|\boldsymbol{\mu}_{ih}, \boldsymbol{\Sigma}_{ih}, \nu_{ih})} \quad \textbf{(17)}$$

in the case of Student's-*t* mixture emissions.

Regarding the posterior expectations of the precision scalars u_{imt} (if applicable), we have

$$\hat{u}_{imt} \triangleq E_{\Theta}(u_{imt}|\mathbf{o}_t) = \frac{\nu_{im} + D}{\nu_{im} + d(\mathbf{o}_t, \boldsymbol{\mu}_{im}; \boldsymbol{\Sigma}_{im})} \quad \textbf{(18)}$$

Further, to obtain the rest of the sought posteriors, we need to define a set of auxiliary distributions, which can be computed by means of a variant of the well-known forward-backward algorithm [83; 4]. Specifically, let us define the forward probabilities

$$\alpha_t(i, k) \triangleq p(\{\mathbf{o}_\tau\}_{\tau=1}^{t}; q_t = i|z_t = k) \quad \textbf{(19)}$$

These probabilities can be computed iteratively, with initialization

$$\alpha_1(i, k) = \begin{cases} \varpi_i p(\mathbf{o}_1|q_1 = i), & k = 1 \\ 0, & k > 1 \end{cases} \quad \textbf{(20)}$$

and recursion

$$\alpha_t(j, k) = p(\mathbf{o}_t|q_t = j)\sum_i \sum_\lambda \pi_{ij}{}^k \alpha_{t-k}(i, \lambda) \quad \textbf{(21)}$$

In a similar way we define the backward probabilities of our model, which yield

$$\beta_t(i, k) \triangleq p(\{\mathbf{o}_\tau\}_{\tau = t+1}^{T} | q_t = i; z_{t+k} = k) \quad \textbf{(22)}$$

These probabilities can also be computed iteratively, with initialization

$$\beta_T(i, k) = 1, \quad \forall k \quad \textbf{(23)}$$

and recursion

$$\beta_t(i, k) = \sum_{j=1}^{M} \sum_\lambda \pi_{ij}^{k} p(\mathbf{o}_{t+k} | q_{t+k} = j) \beta_{t+k}(j, \lambda) \quad \textbf{(24)}$$

Having obtained the forward and backward probabilities of our model, we can now proceed to obtain the remaining sought posteriors. For the emitting state posteriors, hereafter denoted as γ_{jt}, we have

$$\gamma_{jt} \triangleq p(q_t = j | O) \propto \left[\sum_{k=1}^{K} \tilde{\zeta}_{kt} \alpha_t(j, k)\right] \left[\sum_{k'=1}^{K} \tilde{\zeta}_{k', t+k'} \beta_t(j, k')\right] \quad \textbf{(25)}$$

Similarly, the emitting state transition posteriors yield

$$\begin{aligned} \gamma_{ijt}^{\lambda} &\triangleq p(q_t = i, q_{t+\lambda} = j | z_{t+\lambda} = \lambda; O) \quad \textbf{(26)} \\ &\propto \quad \sum_{k, k'=1}^{K} \alpha_t(i, k) \beta_{t+\lambda}(j, k') \pi_{ij}^{\lambda} p(\mathbf{o}_{t+\lambda} | q_{t+\lambda} = j) \end{aligned}$$

Finally, regarding the ("active") dependence form posteriors, hereafter denoted as ζ_{kt}, we have

$$\zeta_{kt} \triangleq E(z_t = k | O) \propto \sum_{i=1}^{N} \alpha_t(i, k) \sum_\lambda \zeta_{\lambda, t+\lambda} \beta_t(i, \lambda) \quad \textbf{(27)}$$

This concludes the E-step of our algorithm.

4.2.3.2 M-step

Having obtained the required posterior expectation expressions on the E-step of the training algorithm of our model, we now proceed to optimization of the objective function (11) over the model parameters to obtain the expressions of the model parameter updates. Let us introduce the notation

$$r_{imt} \triangleq \gamma_{it} \xi_{imt} \qquad \textbf{(28)}$$

We then have

$$\pi_i = \gamma_{i1} \qquad \textbf{(29)}$$

$$\pi_{hi}^{\lambda} = \frac{\sum_t \gamma_{hit}^{\lambda}}{\sum_t \gamma_{ht}} \qquad \textbf{(30)}$$

$$w_{im} = \frac{\sum_{t=1}^{T} r_{imt}}{\sum_{t=1}^{T} \gamma_{it}} \qquad \textbf{(31)}$$

Further, the parameters of the emission distributions yield the following expressions: (i) In case of Gaussian mixture emissions, we have

$$\boldsymbol{\mu}_{im} = \frac{\sum_{t=1}^{T} r_{imt} \mathbf{o}_t}{\sum_{t=1}^{T} r_{imt}} \qquad \textbf{(32)}$$

$$\boldsymbol{\Sigma}_{im} = \frac{\sum_{t=1}^{T} r_{imt} (\mathbf{o}_t - \boldsymbol{\mu}_{im})(\mathbf{o}_t - \boldsymbol{\mu}_{im})^T}{\sum_{t=1}^{T} r_{imt}} \qquad \textbf{(33)}$$

(ii) In case of Student's-*t* mixture emissions, we have

$$\boldsymbol{\mu}_{im} = \frac{\sum_{t=1}^{T} r_{imt}\hat{u}_{imt}\mathbf{o}_t}{\sum_{t=1}^{T} r_{imt}\hat{u}_{imt}} \qquad (34)$$

$$\boldsymbol{\Sigma}_{im} = \frac{\sum_{t=1}^{T} r_{imt}\hat{u}_{imt}(\mathbf{o}_t - \boldsymbol{\mu}_{im})(\mathbf{o}_t - \boldsymbol{\mu}_{im})^T}{\sum_{t=1}^{T} r_{imt}} \qquad (35)$$

while the degrees of freedom are obtained by solving w.r.t. v_{im} the equation

$$\begin{aligned} &1 - \psi\left(\frac{v_{im}}{2}\right) + \log\left(\frac{v_{im}}{2}\right) \\ &+ \psi\left(\frac{\hat{v}_{im} + D}{2}\right) - \log\left(\frac{\hat{v}_{im} + D}{2}\right) \\ &+ \frac{1}{\sum_{t=1}^{T} r_{imt}} \sum_{t=1}^{T} r_{imt}\left(\log\hat{u}_{imt} - \hat{u}_{imt}\right) = 0 \end{aligned} \qquad (36)$$

where $\hat{v}_{im}$ is the current estimate of the degrees of freedom v_{im}, and $\psi(\cdot)$ is the Digamma function.

This concludes the training algorithm of our model. An outline of the EM algorithm for VDJ-HMM is provided in Alg. 1.

EM ALGORITHM FOR THE VDJ-HMM MODEL. ALGORITHM 1

Initialize the model parameters estimate Θ'. Set the maximum number of iterations, $MAXITER$, and the convergence threshold of the EM algorithm.

For $MAXITER$ iterations or until convergence of the objective function $Q(\Theta;\Theta')$**do**:

1. Conduct the forward-backward algorithm to obtain the forward probabilities $\alpha_t(j, k)$ and the backward probabilities $\beta_t(i, k)$, using Eqs. (20)-(21) and (23)-(24), respectively.

2. Effect the E-step of the algorithm by computing the posteriors pertaining to the mixture components, ξ_{imt}, the precision scalars, $\hat{u}_{imt}$, the chain of observation-emitting states, γ_{jt} and $\gamma_{ijt}{}^{\lambda}$, and the Markov chain of dependence jumps, ζ_{kt}. For this purpose, use Eqs. (15), (18), (25)-(26), and (27), respectively.
3. Effect the M-step by computing the new estimates of the model parameters π_i, $\pi_{hi}{}^{\lambda}$, w_{im}, $\boldsymbol{\mu}_{im}$, $\boldsymbol{\Sigma}_{im}$, and ν_{im}, using Eqs. (29)-(36), respectively.

4.2.4 Inference Algorithm

A first inference problem we consider in this work is the problem of predicting the next emitting state, say at time t + 1, denoted as q_{t+1}, given the values of the currently observed data, i.e. the observations set $\{\mathbf{o}_\tau\}_{\tau=1}{}^t$. From the definition of our model, it is easy to deduce that the probability of the emitting state at time t + 1, given the sequence of past observations $\{\mathbf{o}_\tau\}_{\tau=1}{}^t$, can be written in the form

$$\begin{aligned} p(q_{t+1}=j|\{\boldsymbol{o}_\tau\}_{\tau=1}^t) =& \sum_k \sum_{i=1}^N p(q_{t-k+1}=i|\{\boldsymbol{o}_\tau\}_{\tau=1}^t) \\ & \times p(q_{t+1}=j|q_{t-k+1}=i;z_{t-k+1}=k) \\ =& \sum_k \sum_{i=1}^N \pi_{ij}^k \gamma_{i,t-k+1} \end{aligned} \qquad (37)$$

where the emitting state posteriors γ_{jt} are computed by (25), using the sequence of observations $\{\mathbf{o}_\tau\}_{\tau=1}{}^t$. On this basis, determination of the first-layer state of our model, say $\hat{q}$, that is most likely to emit the (next) observation at time t + 1 can be performed by maximization of the conditionals $p(q_{t+1}=j|\{\mathbf{o}_\tau\}_{\tau=1}{}^t)$, yielding:

$$\hat{q} \triangleq \underset{j}{\operatorname{argmax}}\, p(q_{t+1}=j|\{\boldsymbol{o}_\tau\}_{\tau=1}^t) \qquad (38)$$

Another inference problem quite common in the related literature is the task of determining the probability of a given sequence w.r.t. a trained VDJ-HMM model. For this purpose, we can resort to the forward algorithm of our model, similar to conventional HMMs. Specifically, let us consider a sequence $O = \{\mathbf{o}_t\}_{t=1}^{T}$ and a trained VDJ-HMM model with parameter estimate $\hat{\Theta}$. Then, following the definition of our model, the probability of sequence O w.r.t. the available VDJ-HMM model yields

$$p(O|\hat{\Theta}) = \sum_{i=1}^{N} \sum_{k=1}^{K} \alpha_T(i,k) \qquad (39)$$

Finally, as discussed in the Introduction, the key *inference* problem we focus on in this work is the problem of *sequence prediction.* Let us consider a sequence $\{\mathbf{o}_\tau\}_{\tau=1}^{t}$. Then, the sequence prediction problem we consider here is the problem of performing an one-step ahead forecast, i.e. predicting the observation value $\mathbf{o}_{t+1}$ at time $t + 1$, given the values $\{\mathbf{o}_\tau\}_{\tau=1}^{t}$. To address this problem, we exploit the above obtained results regarding computation of the next-state probabilities, $p(q_{t+1} = j|\{\mathbf{o}_\tau\}_{\tau=1}^{t})$. Specifically, we effect the sequence prediction task at time $t + 1$ as follows:

(i) We use Eq. (38) to obtain the emitting state probabilities at the following time point ($t + 1$), given the current set of observations (up to time t), i.e. $p(q_{t+1} = j|\{\mathbf{o}_\tau\}_{\tau=1}^{t})$.

(ii) We set the generated predicted value $\hat{\boldsymbol{o}}_{t+1}$ of the observation at time $t + 1$ equal to the mean value of the modeled variable $\mathbf{o}$ at time $t + 1$, based on the fitted VDJ-HMM model with parameters set $\hat{\Theta}$. Specifically, considering mixtures of Gaussians or Student's-*t* densities as the emission distributions of our model, as discussed previously, this procedure yields:

$$\hat{\boldsymbol{o}}_{t+1} = \sum_{n=1}^{N} \sum_{m=1}^{M} p(q_{t+1} = n | \{\boldsymbol{o}_\tau\}_{\tau=1}^{t}) w_{nm} \boldsymbol{\mu}_{nm} \qquad (40)$$

4.2.5 Computational Complexity

We conclude this section with a short discussion on the computational complexity of our model. We first focus on the training algorithm of our model: From Eqs. (19)-(27), we can easily observe that the main difference between VDJ-HMM model training and training of a simple first-order HMMs concerns computation of the set of forward and backward probabilities, $\{\alpha_t(j, k)\}_{t,j,k}$ and $\{\beta_t(j, k)\}_{t,j,k}$, respectively, which are distinct for each possible temporal dependence pattern, $k = 1, \dots, K$. Indeed, turning to each one of the probabilities in these sets, from Eqs. (21) and (24) we observe that their computation imposes computational costs similar to the corresponding quantities (forward and backward probabilities) pertaining to simple first-order HMMs. Thus, the only difference consists in repeating this computation procedure for each postulated temporal dependence pattern, $k = 1, \dots, K$. Given the fact that the dominant computational costs related to these quantities concerns computation of the emitting state density functions, which are shared across the considered temporal dependence patterns, $k = 1, \dots, K$, it is easy to deduce that, with proper algorithm implementation, VDJ-HMM model training imposes only negligible computational overheads compared to conventional HMM formulations.

Similar results can be obtained regarding the computational costs of inference using our model. Specifically, from Eqs. (37)-(40) we observe that the extra computational costs of the inference procedures of the VDJ-HMM model compared to conventional first-order HMMs are related to the need of computing different state transition posteriors and different forward and backward probabilities for each postulated temporal dependence pattern $k = 1, \ldots, K$. Thus, inference using our model induces only negligible computational overheads compared to conventional HMM formulations, especially when the number of postulated temporal dependence patterns K is considered to be low (which is usually the case in real-world applications, since a relatively low K value selection typically suffices for optimal model performance).

Finally the suggested model exhibits less number of parameters than a high order HMM thus decreasing computational requirements. The following table describes the number of parameters necessary for each of the following variations:

1. Double Layer Hidden Markov Model with different transition probability matrix for each order K Gaussian Mixture HMM
2. K-order Gaussian Mixture HMM

	VDJ-HMM	**HOGHMM**
State Transition Parameters	N*(N-1)*K	N^K*(N-1)

The model is parsimonious because it involves N*(N-1)*K parameters instead of (N-1)*N^K for the full K-order model. Thus for a model specification of order 3 with 4 states it has 36 parameters, instead of 192 for the full high order model. So this structure is capable of modeling first order, higher order and seasonal effects with significant lower number of parameters. We assume that the parameters necessary for the conditional densities are common to all model variations.

4.3 Experiments

In this section, we perform an extensive evaluation of the proposed VDJ-HMM model. For this purpose, we consider a set of time-series forecasting experiments dealing with real-world applications from the computational finance domain. Specifically, we first consider *volatility forecasting* in financial return series; further, we consider the problem of predicting the future *return values* for a set of considered assets. Broad empirical evidence (see, e.g.[102; 106; 104])has shown that financial return series exhibit variable order non-linear temporal dependencies, as well as dependence jumps, both when it comes to *volatility forecasting* and concerning *future value prediction*. As such, leveraging the merits of our model in the context of these applications is expected to yield a significant performance improvement over the competition.

To provide some comparative results, apart from our method we also evaluate the related HMM$^{\infty}$ model [94], which postulates infinitely-long temporal dependencies at each time point, baseline first-order HMMs, and explicit-duration HSMMs [67]. In addition, we cite the performance of methods yielding the state-of-the-art results in the considered experimental scenarios, as they have been reported in the recent literature. In all cases, to ensure the validity of our comparisons, we perform model training following exactly the same experimental setup as in the case of the papers reporting the cited state-of-the-art results.

Our experimental setup is the following: For each one of the considered applications, we split the available data into a training sample, a validation sample, and a testing sample; we adopt the same splits as the authors of the state-of-the-art methods reported in the literature, to render our performance measurements comparable with these results. We use the available training samples to train multiple VDJ-HMM models with different configurations; specifically, we evaluate models with different maximum allowed numbers of alternative temporal dependence patterns (maximum steps back) K, numbers of emitting states N, and numbers of mixture components per emitting state M. We select the optimal model configuration on the basis of the obtained predictive performances on the available validation samples. Finally, we use the available test samples to obtain the reported performance figures. Similar is the experimental setup we adopt for the considered competitors. In all cases, to alleviate the effect of random model initialization on the reported performance results, we repeat our experiments 10 times, with different model initializations each time, and report average performance figures over these repetitions.

4.3.1 Volatility Forecasting

In this set of experiments, we apply our model to prediction of the volatility in daily returns of financial assets. Consider a modeled asset with price P_t at time t; then, its daily return at time t is defined as the logarithm $r_t \triangleq \log P_t / P_{t-1}$. On this basis, (historic) volatility is defined as the square of the return series r_t^2; as discussed in [110], this ground truth measurement constitutes one of the few consistent ways of volatility measuring. As our performance metric used to evaluate the considered algorithms, we consider the root mean squared error (RMSE) between the model-estimated volatilities and the squared returns of the modeled return series (except for the case of the experiments in Sections 3.1.3 and 3.1.4, where we use the mean square error (MSE) and mean absolute error (MAE), following the approach adopted in the related literature).

4.3.1.1 Euro-United States Dollar exchange rate volatility

Our first experimental scenario regarding volatility forecasting is dealing with the EUR-USD exchange rate time series [111]. Specifically, for the purposes of this experiment, we use data from the period 5/17/2007 – 8/10/2008 as our training set, and data pertaining to the period 9/10/2008 - 2/3/2009 as our validation set. To perform model evaluation, we consider three distinct test samples, pertaining to the periods: 3/3/2009 - 10/12/2009, 10/13/2009 - 5/25/2010, and 5/26/2010 - 12/30/2010, respectively. This way, we allow for evaluating model performance in periods with different levels of inherent volatility in the European economy. In all cases, the evaluated methods are trained using a rolling window of the previous 60 days of returns to make daily volatility forecasts for the following 10 days; we retrain the models every 10 days. In our experiments, all the considered HMM-based methods are evaluated using both Gaussian mixtures and Student's-*t* mixtures as their state-conditional emission distributions. In the case of the HSMM method, we consider Poisson, Negative Binomial, Geometric, and Logarithmic densities for modeling state duration.

Table 1: EUR-USD exchange rate volatility: Optimal VDJ-HMM model configuration.

Parameter	Value
K	4
N	2
M	3

In Table 1, we depict the optimal configuration parameters of our model, obtained by utilizing the available validation set, as described previously. In Tables 2 and 3, we illustrate the obtained performances of the evaluated methods. Note that these results are obtained for optimal model configuration (as determined in the validation set) both in the case of our model and the considered competitors. As we observe, in all cases our VDJ-HMM model yields the best performance among the evaluated methods. In addition, it appears that utilization of Student's-*t* mixture emission distributions yields in most cases only negligible performance improvements over models postulating Gaussian mixture emission distributions. We also observe that the HSMM model yielded best performance when postulating Geometric state duration distributions (we omit the results pertaining to different HSMM model configurations for brevity).

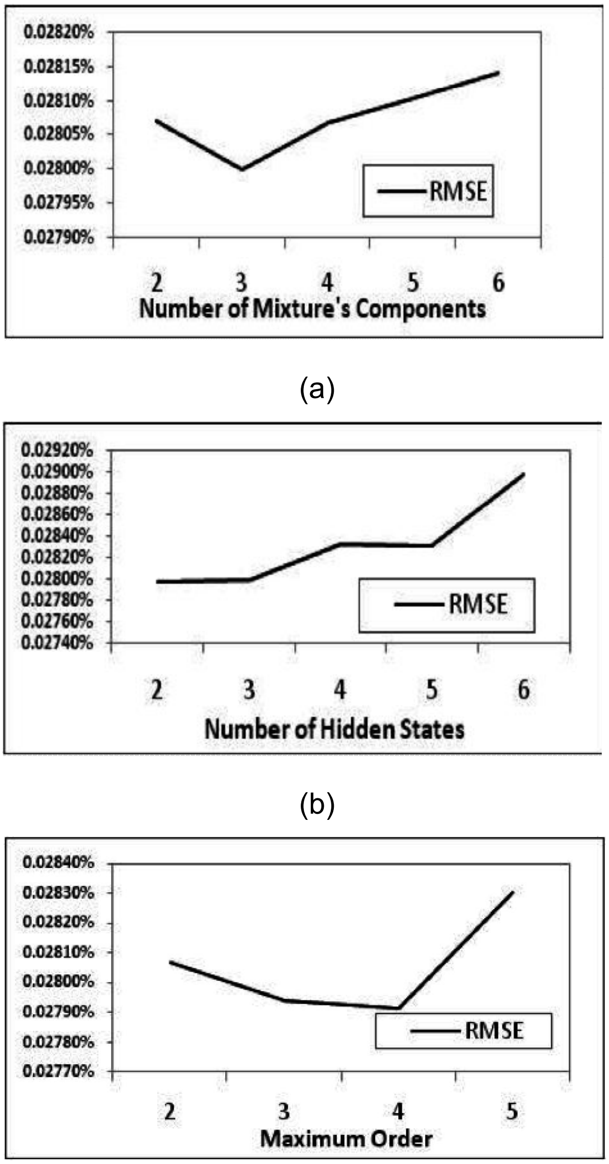
Finally, in Figs 4.2.a-c, we illustrate how model performance changes by varying model configuration, i.e. the hyper parameter values K (maximum order of dependence jumps), N (number of emitting states), and M (number of mixture components). It is apparent that model configuration plays a critical role in the obtained performance. This is especially true for the maximum order of dependence jumps K: selecting too big a value results in performance deterioration, while values close to $K = 1$ (i.e., reducing to a simple first-order HMM) yield inferior performance compared to a fully-fledged VDJ-HMM.

Table 2: EUR-USD exchange rate volatility: Performance (RMSE %) of the evaluated methods.

	HMM	HMM	HMM$^{\infty}$	HMM$^{\infty}$
	(Gaussian)	(Student's-t)	(Gaussian)	(Student's-t)
3/3/2009 - 10/12/2009	1.607	1.559	1.591	1.534
10/13/2009 - 5/25/2010	0.738	0.721	0.730	0.713
5/26/2010 - 12/30/2010	0.683	0.696	0.677	0.691
Total	1.094	1.07	1.086	1.059

Table 3: EUR-USD exchange rate volatility: Performance (RMSE %) of the evaluated methods (cont.).

	HSMM	HSMM	VDJ-HMM	VDJ-HMM
	(Geometric - Gaussian)	(Geometric - Student's-t)	(Gaussian)	(Student's-t)
3/3/2009 - 10/12/2009	1.689	1.74	1.504	1.435
10/13/2009 - 5/25/2010	0.717	0.703	0.7	0.702
5/26/2010 - 12/30/2010	0.681	0.687	0.672	0.669
Total	1.113	1.146	1.028	1.011



(a)

(b)

(c)

Figure 4.2: EUR-USD exchange rate volatility: Performance (RMSE %) fluctuation obtained by varying model configuration (validation set).

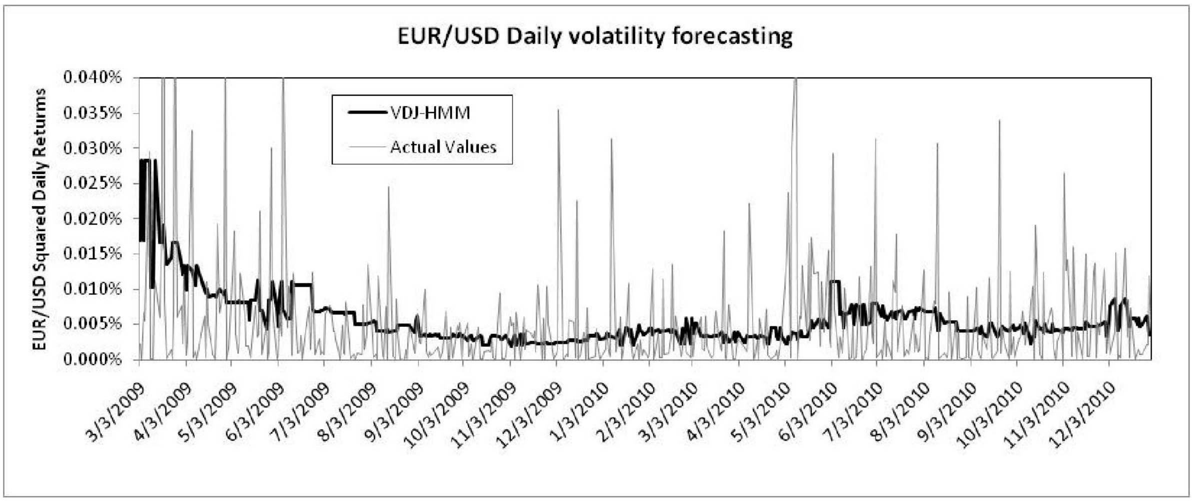


Figure 4.3: EUR/USD daily volatility forecasting graph under the VDJ-HMM

4.3.1.2 Time-series of multiple correlated exchange rates and market indices

In this set of experiments, we consider three application scenarios:

In the first scenario, we model the return series pertaining to the following currency exchange rates, over the period December 31, 1979 to December 31, 1998 (daily closing prices):

1. (AUD) Australian Dollar / US $
2. (GBP) UK Pound / US $
3. (CAD) Canadian Dollar / US $
4. (DKK) Danish Krone / US $
5. (FRF) French Franc / US $
6. (DEM) German Mark / US $
7. (JPY) Japanese Yen / US $
8. (CHF) Swiss Franc / US $.

In the second scenario, we model the return series pertaining to the following *global large-cap equity indices,* for the business days over the period April 27, 1993 to July 14, 2003 (daily closing prices):

1. (TSX) Canadian TSX Composite
2. (CAC) French CAC 40
3. (DAX) German DAX
4. (NIK) Japanese Nikkei 225
5. (FTSE) UK FTSE 100
6. (SP) US S&P 500.

Finally, in the third scenario, we model the return series pertaining to the following seven *global large-cap equity indices* and *Euribor rates,* for the business days over the period February 7, 2001 to April 24, 2006 (daily closing prices for the first 6 indices, and annual percentage rate converted to daily effective yield for the last index):

1. (TSX) Canadian TSX Composite
2. (CAC) French CAC 40
3. (DAX) German DAX
4. (NIK) Japanese Nikkei 225
5. (FTSE) UK FTSE 100
6. (SP) US S&P 500
7. (EB3M) Three-month Euribor rate.

These series have become standard benchmarks for assessing the performance of volatility prediction algorithms[112][113][114]. In our experiments, we follow an evaluation protocol similar to [112; 115].

We adopt the same data split as in [115], all the evaluated methods are trained using a rolling window of the previous 120 days of returns to make daily volatility forecasts for the following 10 days; we retrain the models every 7 days.

To begin with, we consider modeling each asset with a different VDJ-HMM model; i.e. we postulate as many VDJ-HMM models as the assets modeled in each scenario. The same *univariate* setup is also adopted for the considered HMM-based competitors[1]. Under this setup, the determined optimal configuration for our model is provided in Table 4. In Table 5, we provide the obtained results for the three considered scenarios (for optimal model configuration, as determined in the validation set). These results are computed over all the assets modeled in each scenario (averages). The performances of the state-of-the-art methods GARCH [116; 117], mixGARCH [118], VHGP [119], and GPMCH [115]have been cited from [115]. We observe that VDJ-HMM performs better than the competition in all scenarios, with the obtained performance differences becoming more significant in the case of scenario #1, which involves *only* currency exchange rates in the set of modeled assets. We tend to attribute this finding to the fact that currency exchange rates have a unique *mean-reverting property* [120] which seems that our proposed VDJ-HMM model is capable of capturing much better than the competition.

Further, we consider the case of jointly modeling all the assets available in each scenario. For this purpose, we essentially postulate VDJ-HMM models with D-variate emission distributions, where D is the number of jointly modeled assets. The same holds for all the considered HMM-type competitors of our method. In Table 6, we report the determined optimal configuration of our model for this experimental setup. The corresponding predictive performances are reported in Table 7. In this table, we also cite the performance of the multi-output GPMCH model (using Clayton copulas), as reported in [115]. As we observe, our approach yields results comparable to or slightly better than the state-of-the-

[1] All HMM-based models are evaluated using Gaussian mixture emission distributions.

art in all cases. Note also that this performance improvement does also come for a significantly lower computational complexity compared to the second best performing method in these experiments, i.e. the GPMCH method.

Table 4:Time-series of multiple correlated exchange rates and market indices: Optimal VDJ-HMM model configuration under the univariate modeling setup.

Parameter	Value
K	3
N	2
M	2

Table 5:Time-series of multiple correlated exchange rates and market indices: Performance (RMSE %) obtained under the univariate modeling setup.

	HMM	HSMM	HMM∞	GARCH	mixGARCH	VHGP	GPMCH	VDJ-HMM
Scenario #1	0.0442	0.0292	0.0235	0.0705	0.0625	0.0146	0.0121	0.0108
Scenario #2	0.0841	0.0589	0.0353	0.2785	0.2623	0.0552	0.0360	0.0351
Scenario #3	0.0744	0.0578	0.0331	0.0552	0.0550	0.0542	0.0345	0.0329

Table 6:Time-series of multiple correlated exchange rates and market indices: Optimal VDJ-HMM model configuration under the multivariate modeling setup.

Parameter	Value
K	3
N	2
M	2

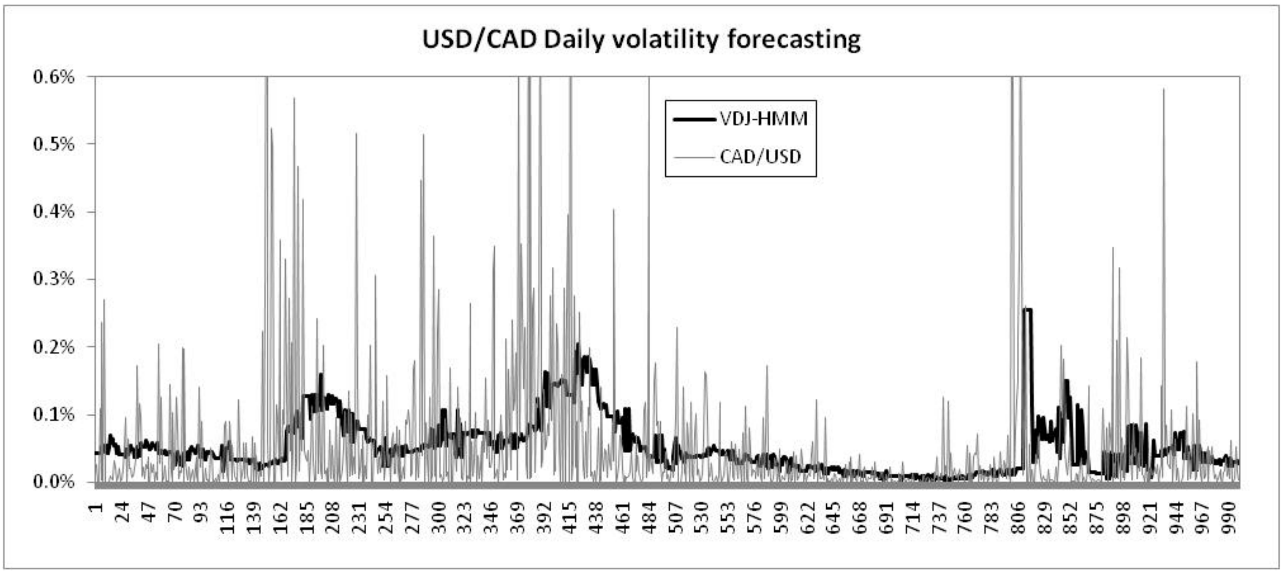


Figure 4.4: Scenario 1 - USD/CAD volatility forecasting graph under the univariate approach

Table 7: Time-series of multiple correlated exchange rates and market indices: Performance (RMSE %) obtained under the multivariate modeling setup.

	HMM	HMM^{∞}	GPMCH	VDJ-HMM
Scenario #1	0.0345	0.0333	0.0341	0.0330
Scenario #2	0.0712	0.0609	0.0557	0.0605
Scenario #3	0.1512	0.1109	0.9905	0.0744

4.3.1.3 Oil price time-series volatility

Further, we consider the problem of volatility forecasting in oil prices. For this purpose, and similar to the experimental setup of [121] we use the daily price data of the Brent index and the West Texas Intermediate (WTI) index from January 6, 1992, to December 31, 2009 (prices expressed in US dollars per barrel). From these time-series, the data pertaining to the last three years, i.e., 2007 to 2009, are used to evaluate the predictive performance of the evaluated models, while the data pertaining to the period 1/3/2006 - 12/29/2006 are used as our validation sample (and the rest for model training). All the evaluated methods are trained using a rolling window of the previous 60 days of returns to make daily volatility forecasts; we retrain the models every 5 days.

In Table 8, we report the optimal configuration of our model for our experiments with both time-series (Brent and WTI). In Table 9, we provide the obtained performances of the evaluated models. Note that all HMM-based models are evaluated using Gaussian mixture emission distributions. The performances of ARCH and its variants have been reported from [121]. As we observe, the proposed VDJ-HMM model consistently yields the best observed performance expressed in terms of the resulting MSE metric, with significant performance differences from all the considered competitors. On the other hand, when evaluation is performed using the MAE metric, we observe that our method manages to yield performance comparable to the state-of-the-art, but it cannot obtain further improvements; note though that the reported state-of-the-art MAEs are already exceptionally low, and therefore the room for further performance improvement is rather limited.

Table 8: Oil price time-series volatility: Optimal VDJ-HMM model configuration.

Parameter	Value (Brent time-series)	Value (WTI time-series)
K	3	3
N	2	2
M	3	4

Table 9: Oil price time-series volatility: Performance (MSE and MAE) of the evaluated approaches.

Method	Brent: MSE	Brent: MAE	WTI: MSE	WTI: MAE
GARCH	0.698	0.065	0.933	0.693
IGARCH	0.856	0.000	0.690	0.000

GJR	0.987	0.811	0.847	0.000
EGARCH	0.609	0.000	0.058	0.000
APARCH	0.557	0.002	0.846	0.031
FIGARCH	0.083	0.111	0.514	0.074
FIAPARCH	0.157	0.586	0.501	0.668
HYGARCH	0.080	0.030	0.546	0.000
HMM	0.087	0.095	0.200	0.067
HSMM	0.100	0.090	0.181	0.090
HMM^{∞}	0.079	0.088	0.191	0.071
VDJ-HMM	0.050	0.001	0.044	0.000

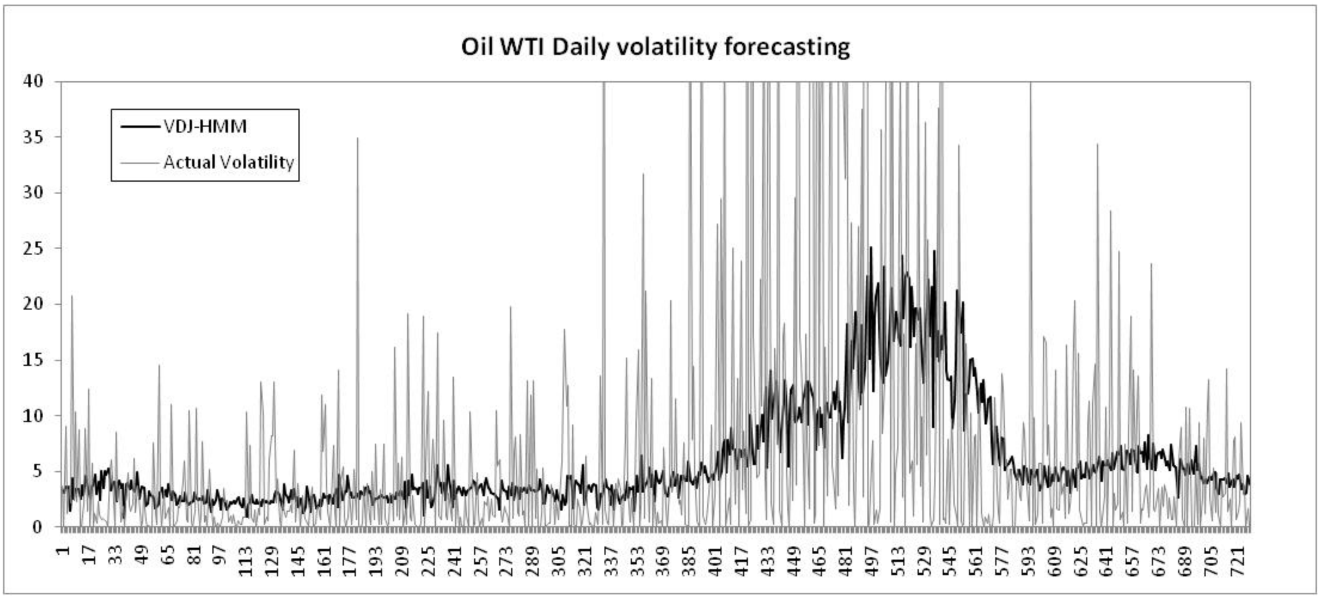


Figure 4.5: Oil WTI daily volatility forecasting graph under the VDJ-HMM

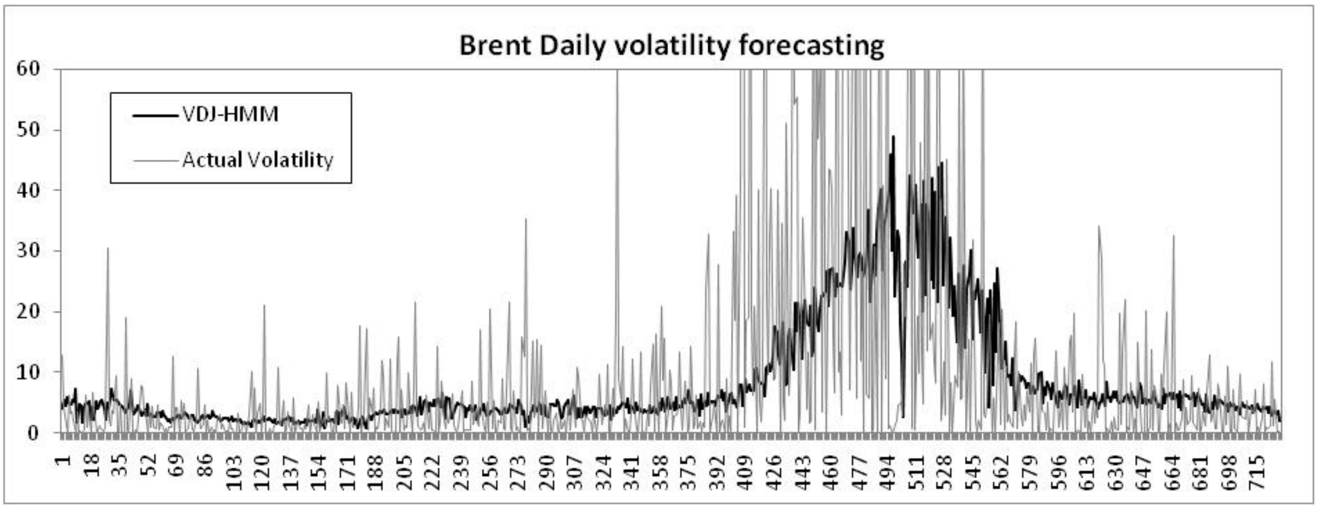


Figure 4.6: Brent daily volatility forecasting graph under the VDJ-HMM

4.3.1.4 Gold market time-series volatility

Finally, we explore the performance of VDJ-HMM in volatility prediction for daily return series of Gold. The dataset used for this experiment consists of the daily Gold fixing prices of the London Bullion Market[2]. Specifically, following [122], we use the daily PM fixings price released at 15:00, and forecast the daily volatility during the second semester of 2008. This is an interesting and quite challenging experimental scenario, since the considered forecast period coincides with the period when the recent financial crisis took place. Similar to [122]our training and validation samples pertain to the period 1/4/1999 - 6/30/2008, while evaluation is performed using the MSE and MAE metrics.

[2] Data obtained from the official website of the London Bullion Market Association (www.lbma.org.uk).

In Table 10, we report the optimal configuration of our model. In Tables 11-12, we provide the obtained performances of the evaluated models. Note that all HMM-based models are evaluated using Gaussian mixture emission distributions. The performances of the reported state-of-the-art competitors, namely historical mean (HM), autoregressive models (AR(k)), moving average models (MA(k) and EWMA), ARMA, as well as several GARCH variants [117; 116], have been cited from [122]. As we observe, the proposed VDJ-HMM model yields a quite satisfactory performance in this experiment, yielding error figures comparable to the state-of-the-art results reported in the recent literature.

Table 10: Gold market time-series volatility: Optimal VDJ-HMM model configuration.

Parameter	Value
K	3
N	3
M	2

Table 11: Gold market time-series volatility: Performance (MSE and MAE) of the evaluated approaches.

	HM	MA(20)	MA(40)	MA(120)	HMM	HSMM	HMM$^{\infty}$
MSE	105.24	84.64	83.29	87.97	85.77	85.5	84.52
MAE	5.43	5.96	5.72	5.40	5.69	5.82	5.63

Table 12:Gold market time-series volatility: Performance (MSE and MAE) of the evaluated approaches (cont.).

	AR(5)	MAD(5)	ARMA	EMWA	GARCH	GARCH-M	VDJ-HMM
MSE	86.08	90.08	84.24	83.81	86.94	86.35	84.16
MAE	5.67	5.54	5.68	5.84	5.56	5.68	5.60

4.3.2 Return Value Prediction

In this set of experiments, we apply our model to prediction of the future values of the daily return series of modeled financial assets, r_t. Specifically, under our experimental setup, we are interested in correctly predicting the sign of the return value at future time points. This sign can be used as the foundation of a simple portfolio management policy as follows: If the predicted future return sign is positive, then the policy suggests that the asset be retained by the investment portfolio manager; on the other hand, if the predicted future return sign is negative, then the policy creates a "sell" signal. All HMM-based models evaluated in these experiments postulate Gaussian mixture models as their emission distributions.

4.3.2.1 Euro-United States Dollar exchange rate

We begin with evaluating our method considering future value prediction for the EUR-USD exchange rate. We use a training sample pertaining to the period 1/17//2002 – 5/16/2008, a validation sample pertaining to the period 5/17/2008 - 3/2/2009, and a test sample pertaining to the period 3/3/2009 - 12/30/2010. All the evaluated methods are trained using a rolling window of the previous 60 days of returns to make daily price prediction for the following 10 days; we retrain the models every 5 days.

On this basis, model evaluation is performed according to: (i) the comparison of the signs of the generated predictions with the actual ones (hereafter referred to as *directional prediction*); and (ii) the resulting annualized return of the aforementioned portfolio management policy, defined as the mean obtained profit adjusted for the return standard deviation over the whole forecasting period.

In Table 13, we depict the optimal configuration of our VDJ-HMM model as determined by utilizing the available validation set. In Figs. a-c, we show how VDJ-HMM model performance changes by varying the adopted configuration (results obtained on the available validation set). As we observe, model configuration plays a crucial role to the obtained performance. Further, another interesting finding is that, similar to the volatility forecasting experiment, model performance reaches its optimal value for a moderate value of K, while experiencing a significant decrease for too high values of K or when $K = 1$.

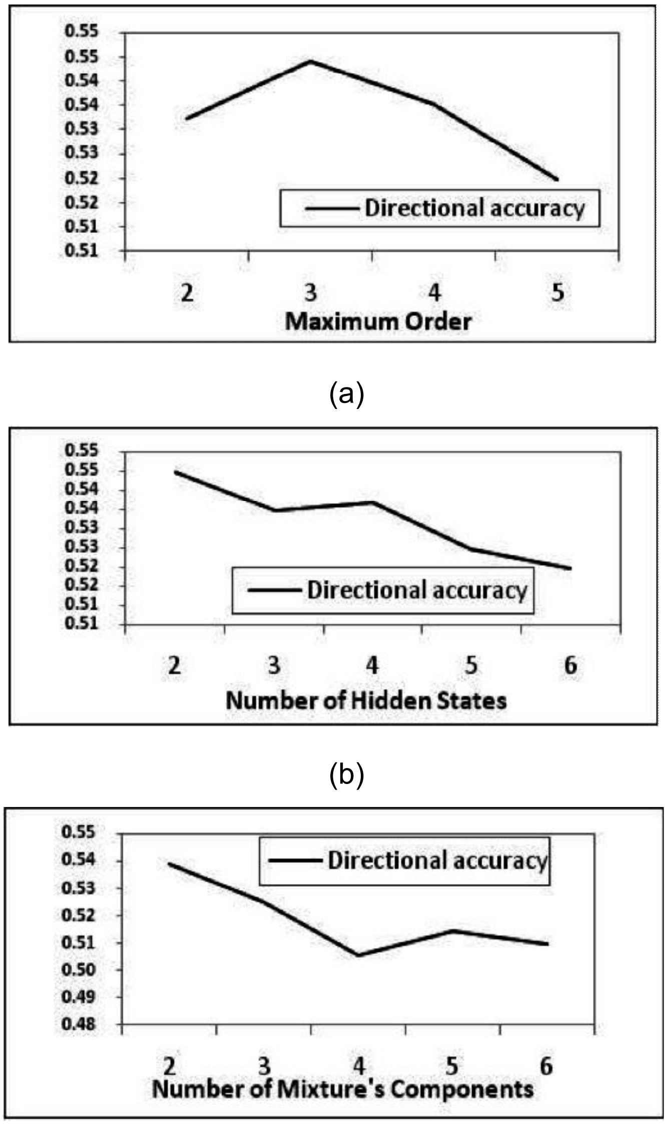
In Table 14, we provide the obtained performance results for the evaluated methods (for optimal model configuration). Note that the performances of the methods k-nearest neighbour (KNN), Naïve Bayes, back-propagation neural network (BP), support vector machine (SVM) [123]and random forest (RF) [124]have been cited from [125]. As we observe, our method completely outperforms the competition, yielding the state-of-the-art result in this dataset.

Table 13: EUR-USD exchange rate price prediction: Optimal VDJ-HMM model configuration.

Parameter	Value
K	3
N	2
M	2

Table 14: EUR-USD exchange rate price prediction: Performance of the evaluated models.

Statistic	KNN	Naïve Bayes	BP	SVM	RF	HMM	HSMM	HMM$^{\infty}$	VDJ-HMM
Directional Prediction Accuracy	50.11%	48.83%	50.12%	52.65%	53.50%	52.5%	51.2%	53.18%	54.05%
Annualized Return	-2.26%	-3.08%	1.59%	3.98%	7.28%	4.05%	1.5%	6.44%	9.50%



(a)

(b)

(c)

Figure 4.7: EUR-USD exchange rate price prediction: Performance fluctuation (directional prediction accuracy) obtained by varying model configuration (validation set).

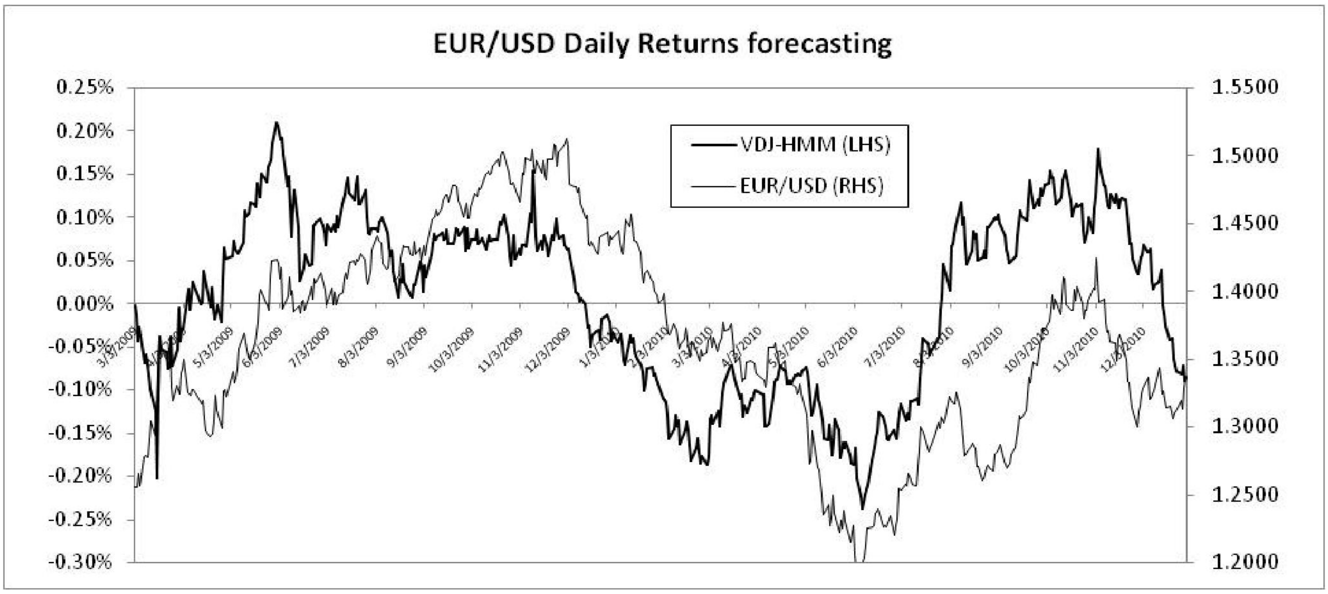


Figure 4.8: EUR-USD exchange rate (right axis) vs return prediction using VDJ-HMM (left axis)

4.3.2.2 Taiwan stock market

Finally, in this experiment, we apply our VDJ-HMM model to predicting the level of the Taiwan stock market (TAIEX). For transparency, in our experimental evaluations we use data pertaining to a seven-year period of the TAIEX, from 1999/1/4 to 2005/12/31, and split them as described in the state-of-the-art work presented in [126]. Specifically, we use the data from the first ten months of each year in the considered time period for model training and validation, and the data from the last two months as our test set. All the evaluated methods are trained using a rolling window of the previous 120 days of returns to make daily price prediction for the following 5 days; we retrain the models every 5 days.

In Table 15, we depict the optimal configuration of our VDJ-HMM model as determined by utilizing the available validation set. In Table 16, we provide the obtained performance figures pertaining to our method and the considered competitors. Performance evaluation is conducted on the grounds of the resulting RMSE between the predicted prices and the actual ones. Note that, apart from HMM and HMM$^{\infty}$, performance of the rest of the considered competitors is cited from [126]. Based on the reported RMSE results, we deduce that VDJ-HMM achieves a clearly competitive performance compared to existing state-of-the-art approaches. Specifically, in some years our VDJ-HMM model achieves the lowest RMSE among the considered competitors, while retaining a very competitive performance in the rest of the examined years.

Table 15: Taiwan stock market price prediction: Optimal VDJ-HMM model configuration.

Parameter	Value
K	3
N	2
M	4

Table 16: Taiwan stock market price prediction: Performance (RMSE) of the evaluated models.

Models	1999	2000	2001	2002	2003	2004	2005
Linear regression	164	420	1070	116	329	146	-
Method of [127]	-	139	144	82	73	-	-
Method of [128]	109	152	130	84	56	79	69
Method of [129]	-	-	122	94	55	69	65
Fuzzy Symmetric Method [126]	103	130	120	68	55	56	54
Fuzzy Asymmetric Method [126]	109	122	125	68	58	58	53
HMM	112	154	116	71	60	59	56
HSMM	111	155	120	75	61	58	59
HMM$^{\infty}$	109	148	113	70	56	58	54
VDJ-HMM	109	145	111	68	55	55	54

Chapter 5 A Novel Corporate Credit Rating System Based on Student's-t Hidden Markov Models

5.1 Introduction

In this work, we focus on the problem of credit scoring/rating of *individual corporations*. In general, a credit scoring/rating system makes use of a statistical technique that combines and analyzes a series of account statement data to predict the future behaviour of a company in terms of its ability to service its debt. The used account data are usually in the form of financial ratios, while the system-generated predictions are typically quantified as the likelihood of occurrence of a default event at some specific future time point.

Credit rating systems are extensively used by the financial sector for the purpose of predicting the evolution of the financial behaviour of an obligor. Reliable prediction of the future behaviour of corporations and measurement of their performance is crucial to private investors for pricing and evaluating their alternative investment options. Financial institutions have been making extensive use of scoring systems for many decades, as a prudent lending practice that allows for better pricing their loan products, and properly quantifying the credit risk embedded in their loan portfolios. In addition, procedures targeted to credit portfolio quality improvement and management of credit losses in the delinquent part of portfolios are highly dependent upon credit scoring systems, which lay the foundation of best practice policies established among financial institutions. As such, credit rating systems are nowadays attached to a series of important internal processes in the financial sector, like pricing, loan granting, provisioning, and risk management.

The introduction of the Basel II framework [130] and its continuation in Basel III [131] has triggered renewed interest in credit rating models research. Specifically, under these frameworks, financial institutions have been granted with the right to develop rating systems for measurement of expected and unexpected losses, allowing for them to establish more risk-sensitive capital adequacy policies. However, Basel II and III frameworks impose specific technical constraints (accuracy requirements) that the used credit rating systems have to comply with. As such, since the introduction of Basel II, banks have been consistently motivated to develop more accurate and robust prediction systems, exploring new statistical techniques especially from the field of statistical machine learning. At the same time, the emergence of the internal ratings-based (IRB) framework has imposed increased requirements on financial institutions regarding collection of financial data from their clients [132]. As such, financial institutions have nowadays accumulated vast amounts of financial ratio data, which can be leveraged by financial researchers so as to develop advanced corporate credit scoring systems. These advances have resulted in a significant enhancement of the sophisticated internal models employed by financial institutions in recent years. This achievement has in turn facilitated a significant decrease in the reliance of the credit approval and risk monitoring internal processes of financial institutions on subjective expert judgment, and has heralded a new era of advanced quantitative reasoning based on objective predictive modeling techniques.

In the last decades, a plethora of alternative approaches have been developed to address the problem of modeling the credit quality of a company, using both quantitative information (e.g., account statements) and qualitative information (e.g., other underwriting criteria, such as obligors market and sector indicators). A first category of approaches belongs to the family of classical regression techniques. [133] used multiple linear discriminant analysis (LDA) to build a rating system for predicting corporate bankruptcies. They estimated a linear discriminant function using liquidity, profitability, leverage, solvency, and turnover financial ratios to estimate credit quality; they dubbed their approach as the Z-score model. One of the main drawbacks of this approach is its assumption that the modeled variables are normally distributed, which is hardly ever the case in real-world scenarios. As such, this method cannot effectively capture nonlinear relationships among the modeled variables, which is crucial for the performance of the credit rating system. In a similar vein, several studies have explored the utility of probit models (e.g, [134]) and linear regression models (e.g., [135]). However, these models continue to suffer from the same drawbacks that plague LDA, namely their clear inability to capture non-linear dynamics, which are prevalent in financial ratio data [136].

Logistic regression is another approach broadly used for building corporate rating systems. It was first used by [137] to predict corporate bankruptcy based on publicly available financial data pertaining to several enterprises (e.g., financial ratios). Logistic regression models employed in this context are essentially used to classify corporations into two distinct classes characterizing their credit risk (i.e., good or bad). Typically, a sigmoid likelihood function is used for modeling purposes to allow for capturing non-linearities and relaxing the normality assumption during model estimation [138].

Decision trees comprise a further category of non-parametric methods used for developing credit rating systems. Decision trees are models that consist of a set of nodes, corresponding to the modeled explanatory variables, and split conditions based on a hierarchical selection of the modeled explanatory variables. Two well-known algorithms in this field are the Chi-squared Automatic Interaction Detector (CHAID) [139] and CART [140] techniques. Decision trees offer simplicity and flexibility in the employed modeling assumptions, while also allowing for easy visualization of the learned modeling strategies (obtained after training). On the negative side, the entailed variable discretization performed by these models results in potential loss of significant information, as well as overfitting proneness.

Another popular class of statistical models used for credit rating is hazard rate models. These models extend the time horizon of a rating system, by looking at the probability of default during the life cycle of the examined loan or portfolio [141; 142]. To achieve this, hazard models explicitly model a survival function for the behaviour of an examined borrower. Cox Proportional hazard models are one popular instance of this type of models [143]; it is based on the assumption that the covariates affecting the default rate are multiplicatively related to the hazard rate function [144].

Finally, research in the field of corporate credit rating has also focused on structural models and reduced form models [145]. Both types of models typically perform predictions on a continuous time frame. Structural models make proper assumptions about the dynamics of a firm's assets and the conditions under which a default may occur. For instance, in this context, [146] used the option-theoretical Black-Sholes model [147] to price default risk. According to the assumptions of [146], a company defaults when the asset value modeled through equity value drops below the debt of the firm. In general, structural models are considered to be more forward-looking and reliable due to their taking into account of market exchange variables. Nevertheless, a problem with such methods is their requirement of availability of firm equity data to perform training, which is not an easy task when dealing with non-listed private or SME companies. On the other hand, the fact that these models take into account market volatility to perform prediction may result in overestimation of default probabilities. In addition, their use of market capitalization to estimate a firm's asset value may lead to discrepancies in information accuracy. Reduced form methods resolve these issues by modeling bankruptcy as a statistical process without making any explicit assumptions as to why a default occurs. Robert Jarrow and Stuart Turnbull proposed in [148]one of the most well known types of reduced-form models for pricing credit risk. This model utilizes multi-factor and dynamic analysis of the term structure of risk-free interest rates, along with martingale theory, to calculate the probability of default.
In this work, we consider following a completely different paradigm towards corporate credit rating. Specifically, we consider using techniques inspired from the literature on machine learning. Indeed, methods from the area of machine learning have been already shown to enhance the capabilities of

conventional corporate credit scoring systems in several studies (e.g., [149]). Among such works, feedforward neural networks (FNNs) constitute the most commonly used machine learning method in the context of corporate credit rating systems [150],[151]. Their successful application in the context of corporate credit rating is basically due to their nonlinear and non-Gaussian modeling assumptions, and their capability to capture dependencies between assets. On the negative side, the notorious proneness of FNNs to overfitting (and, thus, their limited generalization capacity), their need of tedious cross-validation to perform hyperparameter selection (e.g., network size selection), along with their black-box nature that hinders intuitive visualization of the obtained results, limit their potential appeal to the financial community. Other researchers have considered using support vector machines (SVMs) [123]to effect the credit rating task. Indeed, a significant number of studies published in the last decade have shown that SVMs outperform FNNs in credit rating scenarios [152; 153; 154; 155; 156; 157], while reducing the possibility of overfitting, and alleviating the need of tedious cross-validation for the purpose of appropriate hyperparameter selection. On the negative side, SVMs also constitute black-box models, thus limiting their potential of offering deeper intuitions and visualizations regarding the obtained results of their modeling and inference procedure. A Bayesian inference-based analogous to SVMs, namely Gaussian processes, have also been considered by [158]. A drawback of this approach is its high computational complexity, which is cubic to the number of available data points, combined with the assumption of normally distributed data, which is clearly unrealistic, as we have already explained. Finally, Random Forests (RFs) is another type of methods that has recently garnered attention by researchers working in the field of corporate credit rating. This sophisticated technique was introduced in [124], while one successful application of RFs to the problem of corporate credit rating can be found in [159].

Contrary to the above-summarized existing work, in this chapter we propose a novel holistic corporate credit scoring system, that addresses all the parts of the modeling pipeline, from financial ratio time-series selection and pre-processing, to selection of appropriate time-series modeling techniques, and information fusion strategies used to obtain the final credit scores. At the heart of the proposed system lies a novel financial data modeling scheme based on *Student's-t hidden Markov models (SHMMs)* [109]. SHMMs are a successful machine learning technique for modeling data with *temporal dynamics* (i.e., time-series data), that may contain a number of *outliers* and related artefacts in the available training datasets. As such, SHMMs arise as a natural selection for effecting the task of modeling financial ratio data, which entail strong temporal dependencies, while also being quite likely to comprise significant proportions of outliers. Note that this *key* modeling selection of our approach is in *stark* contrast to the machine learning methods used in the context of existing corporate credit scoring systems: existing approaches are based on machine learning models that neither are capable of capturing temporal dependencies in the modeled data, nor can effectively handle outliers in their training datasets. Due to these significant modeling advantages, our approach is expected to yield much better discriminative performance compared to existing alternatives in real-world modeling and prediction scenarios.

Our approach constitutes an intricate data processing pipeline, which comprises a data pre-processing and transformation stage, and a core modeling stage, where SHMMs are used to capture salient temporal patterns in the modeled time-series that are associated with different credit risk scores. To perform modeling and prediction, our approach utilizes appropriate financial ratio time-series, based on the assumption that financial ratios carry all the information necessary to describe and predict the internal state of a company. Specifically, we use five-year historical data of financial ratios, that provide adequate insights on how profitable an examined company is, what the trends are, and how much risk is embedded in its business models. We fit distinct SHMMs to each one of the modeled financial ratios, and obtain separate credit scores from each one of them. Eventually, we train one final information fusion layer that combines the outputs of the individual SHMMs under a weighted linear combination scheme, to generate the final predictions obtained by our system. Parameter optimization of this final information fusion layer is performed by means of a simple yet effective genetic algorithm (GA) [160]. The remainder of the chapter is organized as follows. In Section 2, we provide a brief overview of related work dealing with applications of hidden Markov models (HMMs) to credit risk prediction, and explain the differences between our novel approach and the existing corpus of works. In Section 3, we provide a concise introduction to HMMs, focusing on the case of SHMMs and their training and inference algorithms under the maximum-likelihood framework. In Section 4, we introduce our proposed system, and elaborate on the used data selection and pre-processing schemes, the adopted modeling assumptions and strategies, and the associated training and inference algorithms. In Section 5, we perform the experimental evaluation of our approach: Initially, we elaborate on our experimental setup, and provide details regarding our implementation of the considered alternative methods that we evaluate in parallel to our approach (for comparative purposes). Further, we present our empirical results, analyzing the performance of our proposed SHMM-based corporate credit scoring model, and comparing its performance to the considered state-of-the-art competitors. Finally, in the concluding section, we highlight the performance advantages of our approach; we outline possible limitations of our framework, and discuss areas for future enhancements and research.

5.2 Existing Applications of HMMs to Credit Risk Assessment

HMMs constitute a rather popular method in financial literature. However, up to now their applications to risk assessment have been mostly limited to quantifying risk on *portfolio* level, as opposed to *individual company* level, which is the aim of this work. Specifically, HMMs first appeared in the financial literature in [161], therein, the authors use HMMs to model rating migration of corporate *bonds*, a factor that affects pricing interest rate margins and subsequently the fair value of corporate *bonds*. Further,[162]used HMMs to predict default events in a corporate *portfolio*. Under this approach, the hidden states of the postulated HMMs reflect the state of the economy, which can switch between expansion and recession periods (high risk, normal risk), while the emission distributions of each state are taken as binomial distributions modeling the number of defaults in the studied portfolio at a specific point in time. In [63]the aforementioned model is extended to include exogenous variables (*covariates*), such as interest rates and GDP. More recently, [78]employed an interactive HMM to model corporate *bond* defaults; this model essentially assumes that the relationship between the hidden state of the economy and the evolution of the creditworthiness of the companies in the modeled *portfolio* is *bidirectional*. Finally, the authors of [163] proposed a multistream HMM (MHMM) capable of modeling multiple financial sequences under the assumption that all of them are driven by a common hidden sequence reflecting the state of the economy. They utilize this model to analyze default data in a network of *financial sectors*, and derive reliable estimates of credit value-at-risk (VaR) and expected shortfall for *portfolios* of corporate bonds.

In a different vein, more closely related to our work, [164]proposed an HMM-based model that uses credit ratings posted by rating agencies to perform prediction of the future behaviour of an obligor (default or non-default), where the behaviour variable is modeled as the hidden state of a postulated two-state HMM. More recently, [165]proposed a double-HMM approach which extends the method of [164]by considering as its observed variables both the credit ratings posted by rating agencies and the calculated Altman Z-scores [133]of the examined companies (two streams of information). Even though these approaches are quite closely related to our work, as they are also dealing with *individual corporate credit rating*, there is also a *key* difference that sets them apart from our work: The applicability of the methods in[164; 165], depends on the availability of credit ratings posted by rating agencies, which is hardly the case when dealing with private companies. In contrast, our work does not impose such severely limiting constraints, but offers a bottom-up architecture aiming to obtain reliable corporate credit scores without provision of any prior (expert) information.

5.3 Methodological Background

5.3.1 Student's-t Hidden Markov Models

HMMs are increasingly being adopted in a wide spectrum of applications, since they provide a convenient way of modeling observations appearing in a sequential manner and tending to cluster or to alternate between different possible components (subpopulations) [83]. The observation emission densities associated with each hidden state of a continuous density HMM (CHMM) must be capable of approximating arbitrarily complex probability density functions. Finite Gaussian mixture models (GMMs) are the most common selection of emission distribution models in the CHMM literature, yielding the so-called Gaussian HMMs (GHMMs) [4]. The vast popularity of GHMMs stems from the well-known capability of GMMs to successfully approximate unknown random distributions, including distributions with multiple modes, while also providing a simple and computationally efficient maximum-likelihood (ML) model fitting framework, by means of the expectation-maximization (EM) algorithm [6].Nevertheless, GMMs do also suffer from a significant drawback concerning their parameters estimation procedure, which is well-known that can be adversely affected by the presence of outliers in the data sets used for the model fitting. Hence, when outliers are present in the available fitting data sets (as it often happens in real-world applications), GMMs tend to require excessively high numbers of mixture components to capture the long tails of the approximated distributions (corresponding to the existing outliers), so as to retain their pattern recognition effectiveness. As a consequence of the induced model size increase, the computational efficiency of the trained models deteriorates significantly, while high requirements are also imposed in the size of the available training data sets, so as to guarantee the dependability of the model fitting procedure.

As a solution for the amelioration of these drawbacks, the Student's-*t* HMM (SHMM) has been proposed in [109]as a highly tolerant to outliers alternative to GHMMs. SHMM employs finite mixtures of the longer-tailed multivariate Student's-*t* distribution as its emission distribution models. This selection provides a much more robust approach to data modeling, as training observations that are atypical of a mixture component density are given reduced weight in the calculation of its parameters, under a model-inherent, soundly-founded statistical procedure.

Let us consider a Student's-*t* hidden Markov model comprising I states. Let $\{\mathbf{y}_t\}^T_{t=1}$ denote a sequence of observed data points modeled using the considered SHMM. Let us also assume for convenience, and without any loss of generality, that all the hidden state densities of the considered SHMM are approximated by Student's-*t* mixture models with the *same number* of component distributions, J. Then, from the conditional independence property of the hidden Markov chain [83; 4]it directly follows that the observations emitted from the same hidden state of the SHMM are independent, identically distributed (i.i.d), such that the probability density of the observation $\mathbf{y}_t$ given that it is emitted from the ith model state reads

$$p(\mathbf{y}_t;\mathbf{\Theta}_i) = \Sigma^J_{j=1} c_{ij} t(\mathbf{y}_t;\boldsymbol{\mu}_{ij}, \mathbf{\Sigma}_{ij}, v_{ij}) \qquad \textbf{(1)}$$

where c_{ij}, $\boldsymbol{\mu}_{ij}$, $\mathbf{\Sigma}_{ij}$ and v_{ij} are the mixing proportion, mean, covariance matrix and the degrees of freedom of the jth component density of the hidden distribution of the ith state of the model, respectively, and $\mathbf{\Theta}_i = \{c_{ij}, v_{ij}, \boldsymbol{\mu}_{ij}, \mathbf{\Sigma}_{ij}\}^J_{j=1}$ $(i = 1, ..., I)$. The probability density function (pdf) of a Student's-*t* distribution with mean vector $\boldsymbol{\mu}$, covariance matrix $\mathbf{\Sigma}$, and $v > 0$degrees of freedom is [166]

$$t(\mathbf{y}_t;\boldsymbol{\mu}, \mathbf{\Sigma}, v) = \frac{\Gamma(\frac{v+p}{2})|\mathbf{\Sigma}|^{-1/2}(\pi v)^{-p/2}}{\Gamma(v/2)\{1 + d(\mathbf{y}_t, \boldsymbol{\mu};\mathbf{\Sigma})/v\}^{(v+p)/2}} \qquad \textbf{(2)}$$

where p is the dimensionality of the observations $\mathbf{y}_t$, $d(\mathbf{y}_t, \boldsymbol{\mu};\boldsymbol{\Sigma})$ is the squared Mahalanobis distance between $\mathbf{y}_t$, $\boldsymbol{\mu}$ with covariance matrix $\boldsymbol{\Sigma}$, and $\Gamma(s)$ is the Gamma function,

$$\Gamma(s) = \int_0^\infty e^{-t} z^{s-1} dz.$$

Directed acyclic graph representing the SHMM Figure 5.1 [109].The box (plate) denotes a set of T observation points, $\{\mathbf{y}_{mt}\}^{T}_{t=1}$ (of which only a single example for time t is shown explicitly), with their corresponding previous and current state indicators, and their mixture component indicators.

5.3.2 Model Training

Training of the SHMM using multiple training sequences (tokens) can be easily conducted by means of the expectation-maximization (EM) algorithm. Let us consider M independent sequences of fitting data. We assume for convenience, that all the sequences have the same length T, i.e. they comprise T data points, without any loss of generality. Let the mth sequence be $\mathbf{y}_m = \{\mathbf{y}_{mt}\}^{T}_{t=1}$, $m = 1, \ldots, M$, where $\mathbf{y}_{mt}$ stands for the tth data point of the mth fitting sequence. Then, from (1), we have

$$p(\mathbf{y}_{mt};\boldsymbol{\Theta}_i) = \sum_{j=1}^{J} c_{ij} t(\mathbf{y}_{mt};\boldsymbol{\mu}_{ij}, \boldsymbol{\Sigma}_{ij}, \nu_{ij}) \qquad \textbf{(3)}$$

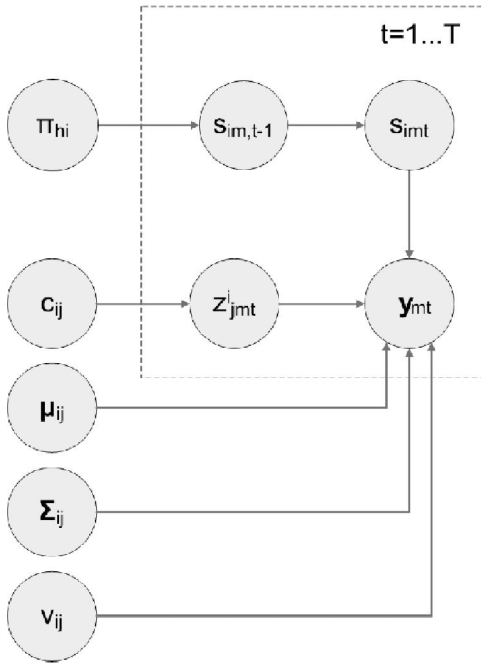


Figure 5.1: Directed acyclic graph representing the SHMM (Chatzis et al., 2009). The box (plate) denotes a set of T observation points, {ymt}Tt=1 (of which only a single example for time t is shown explicitly), with their corresponding previous and current state indicators, and their mixture component indicators.

or, equivalently, using the properties of the Student's-*t* distribution (c.f., [167; 109]):

$$p(\mathbf{y}_{mt}|\{u_{ijmt}\}_{j=1}^{J}; \mathbf{\Theta}_i) = \sum_{j=1}^{J} c_{ij}\mathcal{N}(\mathbf{y}_{mt};\boldsymbol{\mu}_{ij}, \mathbf{\Sigma}_{ij}/u_{ijmt}) \qquad (4)$$

where u_{ijmt} is a precision scalar corresponding to the observation $\mathbf{y}_{mt}$ given it is generated from the *j*th component density of the *i*th hidden state distribution

$$u_{ijmt} \sim \mathcal{G}(\frac{v_{ij}}{2}, \frac{v_{ij}}{2}) \qquad (5)$$

Let us denote as $\mathbf{s}_{mt}$ the state indicator vectors of the observed data, with $\mathbf{s}_{mt} = (s_{imt})^{I}_{i=1}$, and

$$s_{imt} \triangleq \begin{cases} 1, & \text{if } \mathbf{y}_{mt} \text{ is emitted from the } i\text{th model state} \\ 0, & \text{otherwise} \end{cases}$$

Let us also denote as $\mathbf{z}^i_{mt}$ the state-conditional mixture component indicator vectors of the observed data, such that $\mathbf{z}^i_{mt} = (z^i_{jmt})^J_{j=1}$, and, given that $\mathbf{y}_{mt}$ is emitted from the ith state ($s_{imt} = 1$), it holds

$$z^i_{jmt} \triangleq \begin{cases} 1, & \text{if } \mathbf{y}_{mt} \text{ is generated from the } j\text{th component} \\ & \text{density of the state} \\ 0, & \text{otherwise} \end{cases}$$

The EM algorithm comprises optimization of the posterior expectation of the complete data log-likelihood of the treated model

$$Q(\boldsymbol{\Psi};\boldsymbol{\Psi}') \triangleq E_{\boldsymbol{\Psi}'}(\log L_c(\boldsymbol{\Psi})|\mathbf{y}) \qquad \textbf{(6)}$$

where $\boldsymbol{\Psi}'$ denotes the obtained estimator of the model parameters vector $\boldsymbol{\Psi} = \{\boldsymbol{\Theta}_i, \pi_i, \pi_{hi}\}^I_{h,i=1}$, π_i are the initial state probabilities, and π_{hi} are the state transition probabilities of the Markov chain. For a continuous hidden Markov model, the expression of the complete data log-likelihood reads [167]

$$\log L_c(\boldsymbol{\Psi}) = \sum_{m=1}^{M} \sum_{h=1}^{I} \left[s_{hm1} \log \pi_h + \sum_{i=1}^{I} \sum_{t=1}^{T-1} s_{hmt} s_{im,t+1} \log \pi_{hi} \right] \qquad \textbf{(7)}$$

$$+ \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{t=1}^{T} s_{imt} \log p(\mathbf{y}^{comp}_{mt};\boldsymbol{\Theta}_i)$$

where $\mathbf{y}^{comp}{}_{mt}$ stands for the complete data corresponding to the tth observation of the mth sequence, $\mathbf{y}_{mt}$, and $\log p(\mathbf{y}^{comp}{}_{mt};\mathbf{\Theta}_i)$ is the complete data log-likelihood of the emission distribution of the ith hidden state with respect to $\mathbf{y}_{mt}$. A graphical illustration (plate diagram) of the considered SHMM can be found in Fig. 5.1.

To provide a proper complete data configuration for the SHMM, we have to take into account that a closed form solution for log-likelihood optimization of a Student's-t mixture in the form (3) does not exist [166; 167]. However, exploiting the alternative expression (4)-(5) of a Student's-t distribution as a Gaussian distribution with scaled precision, where the scalar is a Gamma distributed latent variable, a tractable optimization framework is obtained. Hence, we let the complete data corresponding to the mth sequence, $\mathbf{y}^{comp}{}_{m}$, comprise the observable data and their corresponding state indicator vectors, state-conditional mixture component indicator vectors, and precision scalars. Then, we have

$$p(\mathbf{y}^{comp}_{mt};\mathbf{\Theta}_i) = \prod_{j=1}^{J} \left[c_{ij} p(\mathbf{y}_{mt}|u_{ijmt};\mathbf{\Theta}_i) p(u_{ijmt};\mathbf{\Theta}_i)\right]^{z^i_{jmt}}$$

which yields (ignoring constant terms)

$$\log p(\mathbf{y}^{comp}_{mt};\mathbf{\Theta}_i) = \sum_{j=1}^{J} z^i_{jmt} \Big\{ -\log\Gamma(\tfrac{\nu_{ij}}{2}) + \tfrac{\nu_{ij}}{2} \times \tag{8}$$

$$\Big[\log(\tfrac{\nu_{ij}}{2}) + \log u_{ijmt} - u_{ijmt}\Big] + \log c_{ij}$$

$$. - \frac{u_{ijmt}}{2} d(\mathbf{y}_{mt}, \boldsymbol{\mu}_{ij};\mathbf{\Sigma}_{ij}) - \frac{1}{2}\log|\mathbf{\Sigma}_{ij}| \Big\}$$

The E-step on the (k+1)th iteration of the EM algorithm requires calculation of the quantity $Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(k)})$, where $\boldsymbol{\Psi}^{(k)}$ denotes the *current* estimator (obtained by the kth iteration of the EM algorithm) of $\boldsymbol{\Psi}$. Using (7) and (8), we have

$$Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(k)}) = \sum_{m=1}^{M} \sum_{h=1}^{I} [\gamma^{(k)}_{hm1} \log\pi_h + \sum_{i=1}^{I} \sum_{t=1}^{T-1} \gamma^{(k)}_{himt} \log\pi_{hi}] \tag{9}$$

$$+ \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma^{(k)}_{imt} E_{\boldsymbol{\Psi}^{(k)}}\left(\log p(\mathbf{y}^{comp}_{mt};\boldsymbol{\Theta}_i)|\mathbf{y}\right)$$

where $\gamma^{(k)}{}_{imt}$ denote the kth iteration estimators of the state emission posterior probabilities, defined as

$$\gamma_{imt} \triangleq p(s_{imt} = 1|\,\mathbf{y}) = p(s_{imt} = 1|\,\mathbf{y}_m) \tag{10}$$

(t = 1, ..., T), and $\gamma^{(k)}{}_{himt}$ denote the kth iteration estimators of the state transition posterior probabilities, defined as

$$\gamma_{himt} \triangleq p(s_{im,\,t+1} = 1,\, s_{hmt} = 1|\,\mathbf{y}) \tag{11}$$

(t = 1, ..., $T-1$) for m = 1, ..., M, h, i = 1, ..., I. Therefore, the E-step of the algorithm comprises computation of the estimates $\gamma^{(k)}{}_{imt}$ and $\gamma^{(k)}{}_{himt}$, and of the expectation$E_{\boldsymbol{\Psi}}{}^{(k)}(\log p(\mathbf{y}^{comp}{}_{mt};\boldsymbol{\Theta}_i)|\mathbf{y})$. Let us begin with the updates $\gamma^{(k)}{}_{imt}$ and $\gamma^{(k)}{}_{himt}$. These quantities can be obtained utilizing the forward-backward algorithm. It holds [4; 83]

$$\gamma^{(k)}_{himt} = \frac{a^{(k)}_{hmt}\pi^{(k)}_{hi}p(\mathbf{y}_{m,\,t+1};\,\mathbf{\Theta}^{(k)}_{i})b^{(k)}_{im,\,t+1}}{\sum_{\upsilon=1}^{I}\sum_{\varphi=1}^{I}a^{(k)}_{\upsilon mt}\pi^{(k)}_{\upsilon\varphi}p(\mathbf{y}_{m,\,t+1};\,\mathbf{\Theta}^{(k)}_{\varphi})b^{(k)}_{\varphi m,\,t+1}} \qquad \textbf{(12)}$$

and

$$\gamma^{(k)}_{imt} = \frac{a^{(k)}_{imt}\,b^{(k)}_{imt}}{\sum_{h=1}^{I}a^{(k)}_{hmt}\,b^{(k)}_{hmt}} \qquad \textbf{(13)}$$

where

$$a^{(k)}_{im1} = \pi^{(k)}_{i}p(\mathbf{y}_{m1};\,\mathbf{\Theta}^{(k)}_{i}) \qquad \textbf{(14)}$$

$$a^{(k)}_{im,\,t+1} = p(\mathbf{y}_{m,\,t+1};\,\mathbf{\Theta}^{(k)}_{i})\sum_{h=1}^{I}a^{(k)}_{hmt}\pi^{(k)}_{hi}\ (t = 1,\,\ldots,\,T-1) \qquad \textbf{(15)}$$

$$b^{(k)}_{hmT} = 1 \qquad \textbf{(16)}$$

$$b^{(k)}_{hmt} = \sum_{i=1}^{I}\pi^{(k)}_{hi}p(\mathbf{y}_{m,\,t+1};\,\mathbf{\Theta}^{(k)}_{i})b_{im,\,t+1}\ (t = T-1,\,\ldots,\,1) \qquad \textbf{(17)}$$

and the expression of $p(\mathbf{y}_{mt};\mathbf{\Theta}_i)$ is given by (3). Concerning the term $E_{\mathbf{\Psi}}{}^{(k)}(\log p(\mathbf{y}^{comp}{}_{mt};\mathbf{\Theta}_i)|\mathbf{y})$, it can be shown that its estimation reduces to computation of the conditional posteriors of mixture component membership

$$\xi^{(k)}_{ijmt} \triangleq E_{\boldsymbol{\Psi}^{(k)}}(z^{i}_{jmt} | \mathbf{y}_{mt}, s_{imt} = 1) = \frac{c^{(k)}_{ij} t(\mathbf{y}_{mt}; \boldsymbol{\mu}^{(k)}_{ij}, \boldsymbol{\Sigma}^{(k)}_{ij}, \nu^{(k)}_{ij})}{\sum_{h=1}^{J} c^{(k)}_{ih} t(\mathbf{y}_{mt}; \boldsymbol{\mu}^{(k)}_{ih}, \boldsymbol{\Sigma}^{(k)}_{ih}, \nu^{(k)}_{ih})} \qquad \textbf{(18)}$$

and of the posterior expectations of the precision scalars u_{ijmt}

$$u^{(k)}_{ijmt} \triangleq E_{\boldsymbol{\Psi}^{(k)}}(u_{ijmt} | \boldsymbol{y}_{mt}) = \frac{\nu^{(k)}_{ij} + p}{\nu^{(k)}_{ij} + d(\boldsymbol{y}_{mt}, \boldsymbol{\mu}^{(k)}_{ij}; \boldsymbol{\Sigma}^{(k)}_{ij})} \qquad (19)$$

Finally, the M-step of the algorithm is effected by performing the computations

$$\pi^{(k+1)}_{i} = \frac{1}{M} \sum_{m=1}^{M} \gamma^{(k)}_{im1} \qquad \textbf{(20)}$$

$$\pi^{(k+1)}_{hi} = \frac{\sum_{m=1}^{M} \sum_{t=1}^{T-1} \gamma^{(k)}_{himt}}{\sum_{m=1}^{M} \sum_{t=1}^{T} \gamma^{(k)}_{hmt}} \qquad \textbf{(21)}$$

$$c^{(k+1)}_{ij} = \sum_{m=1}^{M} \sum_{t=1}^{T} r^{(k)}_{ijmt} \Big/ \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma^{(k)}_{imt} \qquad \textbf{(22)}$$

$$\boldsymbol{\mu}^{(k+1)}_{ij} = \frac{\sum_{m=1}^{M} \sum_{t=1}^{T} r^{(k)}_{ijmt} u^{(k)}_{ijmt} \mathbf{y}_{mt}}{\sum_{m=1}^{M} \sum_{t=1}^{T} r^{(k)}_{ijmt} u^{(k)}_{ijmt}} \qquad \textbf{(23)}$$

$$\boldsymbol{\Sigma}^{(k+1)}_{ij} = \sum_{m=1}^{M}\sum_{t=1}^{T} r^{(k)}_{ijmt}\, u^{(k)}_{ijmt} (\mathbf{y}_{mt} - \boldsymbol{\mu}^{(k+1)}_{ij})(\mathbf{y}_{mt} - \boldsymbol{\mu}^{(k+1)}_{ij})^{T} \times [\sum_{m=1}^{M}\sum_{t=1}^{T} r^{(k)}_{ijmt}]^{-1} \qquad \textbf{(24)}$$

and solving the equation

$$1 - \psi(\frac{v_{ij}}{2}) + \log(\frac{v_{ij}}{2}) + \psi(\frac{v^{(k)}_{ij} + p}{2}) - \log(\frac{v^{(k)}_{ij} + p}{2}) + \frac{1}{\sum_{m=1}^{M}\sum_{t=1}^{T} r^{(k)}_{ijmt}} \sum_{m=1}^{M}\sum_{t=1}^{T} r^{(k)}_{ijmt} \left(\log u^{(k)}_{ijmt} - u^{(k)}_{ijmt}\right) = 0 \qquad \textbf{(25)}$$

to obtain the estimates of v_{ij}, where, $\psi(s)$ is the digamma function, and $r^{(k)}{}_{ijmt}$ is the joint posterior probability that $\mathbf{y}_{mt}$ is generated from the ith state of the model and particularly from its jth component distribution

$$r_{ijmt} \triangleq p(s_{imt} = 1, z^{i}_{jmt} = 1 \mid \mathbf{y}) = \gamma_{imt}\xi_{ijmt} \qquad \textbf{(26)}$$

$$r^{(k)}_{ijmt} = \gamma^{(k)}_{imt}\xi^{(k)}_{ijmt} \qquad \textbf{(27)}$$

An outline of the EM algorithm for the SHMM is given in Alg. 1.

EM ALGORITHM FOR THE SHMM. ALGORITHM 1

k: = 0

1. Conduct the forward-backward algorithm to obtain the quantities $a^{(k)}{}_{imt}$ and $b^{(k)}{}_{imt}$.

2. Effect the E-step by computing the $\gamma^{(k)}{}_{himt}$, $\gamma^{(k)}{}_{imt}$, $\xi^{(k)}{}_{ijmt}$, $r^{(k)}{}_{ijmt}$, and $u^{(k)}{}_{ijmt}$, using (12), (13), (18), (27), and (19), respectively.
3. Effect the M-step by computing the $\pi^{(k+1)}{}_{i}$, $\pi^{(k+1)}{}_{hi}$, $c^{(k+1)}{}_{ij}$, $\boldsymbol{\mu}^{(k+1)}{}_{ij}$, $\boldsymbol{\Sigma}^{(k+1)}{}_{ij}$, and $v^{(k+1)}{}_{ij}$, using (20)-(25), respectively.
4. If the EM algorithm converges, **exit**; otherwise increase the iteration counter ($k := k + 1$) and goto 1.

5.3.3 Inference Algorithm

Given a trained SHMM, inference using this model consists in calculating the likelihood of a given sequence, and estimating the emitting (hidden) states sequence corresponding to an observed sequence presented to the model. Let us consider an SHMM, trained using the EM algorithm, as described above, with parameters set $\hat{\boldsymbol{\Psi}}$, and an observed sequence $\mathbf{y} = \{\mathbf{y}_t\}^T{}_{t=1}$. Then, likelihood calculation can be performed by utilizing the forward algorithm. Specifically, following [109] the likelihood $p(\mathbf{y}|\hat{\boldsymbol{\Psi}})$ yields

$$p(\boldsymbol{y}|\hat{\boldsymbol{\Psi}}) = \sum_{i=1}^{I} \hat{a}_{i,T} \qquad (28)$$

where $\{\hat{a}_{i,t}\}^{I,T}{}_{i,t=1}$ are the forward probabilities corresponding to the observed sequence $\mathbf{y}$, computed using (14)-(15) and the parameter estimates $\hat{\boldsymbol{\Psi}}$ of the postulated SHMM. On the other hand, the task of estimating the hidden states sequence corresponding to the observed sequence $\mathbf{y}$ can be effected by means of the Viterbi algorithm. Following [109], the estimate of the current hidden state at time t, $\hat{s}_t$, yields

$$\hat{s}_t = \arg\max_{1\leq i\leq I} \delta_t(i) \qquad (29)$$

where

$$\delta_t(j) = \max_{1 \leq i \leq I} \left\{ p(\boldsymbol{x}_t | y_t = j) \pi_{ij} \right\} \delta_{t-1}(i), \ t > 1 \tag{30}$$

with initialization

$$\delta_1(j) = p(\boldsymbol{x}_1 | y_1 = j) \pi_j \tag{31}$$

5.4 Proposed Approach

5.4.1 Research Motivation

As we have already discussed, corporate credit risk modeling and prediction is typically based on modeling appropriate financial ratio data. On this basis, our approach is motivated from some key insights regarding the nature of the modeled data: It is well understood that creditworthiness patterns exhibit strong temporal dependencies that are reflected in, and can be extracted from, financial ratio data [168; 169]. Indeed, corporate credit ratings are well-known to be largely driven by the hidden state of the business cycle process. As such, using machine learning models capable of robustly capturing temporal dynamics in the modeled data is expected to significantly enhance the discriminatory capacity of a developed corporate risk rating system. In addition, outliers and related artifacts are rather common in financial time-series datasets used for model training [170; 171]. Therefore, coming up with a modeling method with training algorithms tolerant to the existence of outliers in the used training data is expected to result in better trained models, with enhanced predictive accuracy.

Under this motivation, in this work we suggest to use SHMMs as the core component of a corporate credit rating system, trained on financial ratio time-series. SHMMs satisfy both our requirements of postulating models capable of capturing temporal dynamics in the modeled data, and using models tolerant to outliers in their training data. This formulation is in *stark* contrast to existing machine learning-based approaches used for corporate credit rating, e.g., approaches based on FNNs and RFs, regression techniques, decision trees, and hazard models, which are not capable of extracting temporal dynamics in the modeled data, and, thus, cannot capture changes in the business cycle that could lead in a significant shift in the behaviour of the modeled businesses. Another significant merit of our approach that sets it apart from existing approaches is that our use of SHMMs affords modeling *continuous* measured variables pertaining to financial ratios. Finally, as we have shown in Section 3, model training and inference for SHMMs can be performed using robust, elegant, and computationally efficient algorithms with proved convergence [109]. It allows for increased robustness to outliers in the training data, and poses no substantial computational overheads compared to existing competitors.

5.4.2 System Architecture

As depicted in Fig. 5.2, the proposed corporate credit scoring system comprises four distinct processing stages, namely: (i) data collection and processing; (ii) SHMM model training for each financial ratio; (iii) model aggregation; and (iv) system calibration. We elaborate on each one of these stages in the remainder of this section.

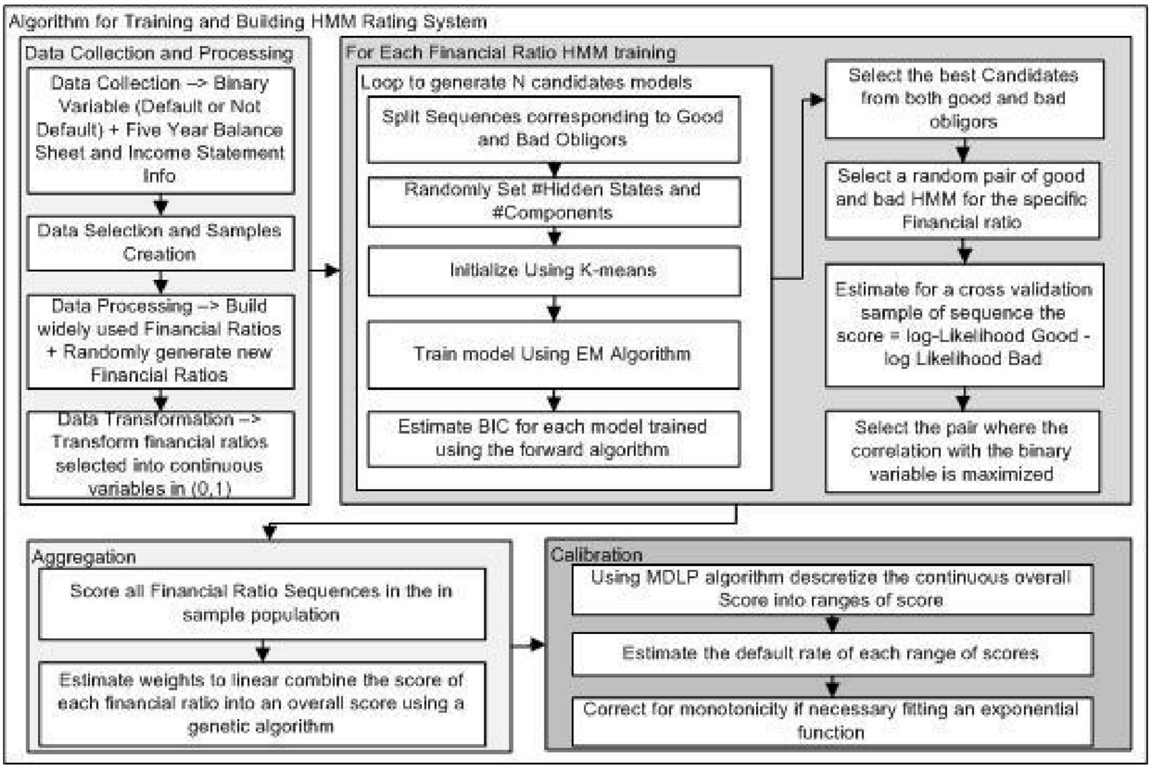

Algorithm for Training and Building HMM Rating System
Data Collection and Processing
Data Collection –> Binary Variable (Default or Not Default) + Five Year Balance Sheet and Income Statement Info
Data Selection and Samples Creation
Data Processing –> Build widely used Financial Ratios + Randomly generate new Financial Ratios
Data Transformation –> Transform financial ratios selected into continuous variables in (0,1)
For Each Financial Ratio HMM training
Loop to generate N candidates models
Split Sequences corresponding to Good and Bad Obligors
Randomly Set #Hidden States and #Components
Initialize Using K-means
Train model Using EM Algorithm
Estimate BIC for each model trained using the forward algorithm
Select the best Candidates from both good and bad obligors
Select a random pair of good and bad HMM for the specific Financial ratio
Estimate for a cross validation sample of sequence the score = log-Likelihood Good - log Likelihood Bad
Select the pair where the correlation with the binary variable is maximized
Aggregation
Score all Financial Ratio Sequences in the in sample population
Estimate weights to linear combine the score of each financial ratio into an overall score using a genetic algorithm
Calibration
Using MDLP algorithm descretize the continuous overall Score into ranges of score
Estimate the default rate of each range of scores
Correct for monotonicity if necessary fitting an exponential function

Figure 5.2: Proposed Rating System Architecture (Training and Building).

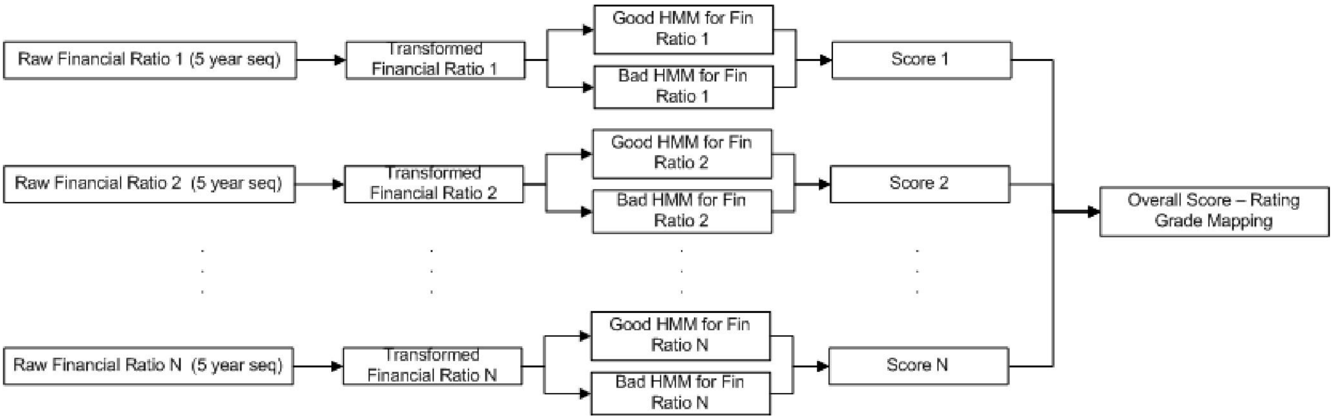

Raw Financial Ratio 1 (5 year seq)
Transformed Financial Ratio 1
Good HMM for Fin Ratio 1
Bad HMM for Fin Ratio 1
Score 1
Raw Financial Ratio 2 (5 year seq)
Transformed Financial Ratio 2
Good HMM for Fin Ratio 2
Bad HMM for Fin Ratio 2
Score 2
Overall Score – Rating Grade Mapping
Raw Financial Ratio N (5 year seq)
Transformed Financial Ratio N
Good HMM for Fin Ratio N
Bad HMM for Fin Ratio N
Score N

Figure 5.3: Proposed rating system calculation workflow.

5.4.2.1 Data Collection and Processing

This comprises data collection, data processing and transformation, and data selection and samples creation.

Data Collection. Training data collection is a significant procedure for the effectiveness of a machine learning system. In the context of our corporate credit rating system, we have collected information on performing and non-performing entities from the supervisory database of the Central Bank of Greece. These data were aggregated during each year from 2006 to 2012, according to the established regulatory framework. The collected information is related with both SMEs and corporations with loans granted from Greek banks; the adopted definition of a default event in this dataset is in line with the rules of Basel III [131]. Specifically, a loan is flagged as delinquent if it is either 90 days past due or it gets rated as delinquent based on each bank's internal rating rules. In the beginning of the observation periods, all considered obligors are performing. In our data collection procedures, we do not consider special cases of obligors from the financial sector, including banks, insurance, leasing, and factoring companies, due to the very unique nature of their business models, which deviate quite a lot from the business models of commercial companies. Under our proposed framework, each obligor is considered to be categorized as either *good* (i.e., performing) or *bad* (i.e., non-performing). Each company was either good or bad at the end of the observation periods. An obligor is categorized as *good* if *at least one of the following criteri*a is met: (i) the obligor manages to get an upgrade of their rating by their bank (according to the bank's internal rules); (ii) the obligor is not delinquent; and (iii) the obligor receives a good rating based on the internal rating system of their bank.

Data Selection and Samples Creation. To create our used datasets, we randomly select good and bad clients from the available population. Each of these clients is represented using a set of financial ratio time-series, extracted using their available balance sheet data and income (P&L) statements. In each case, the dependent variable in our training datasets is a binary indicator, with the on value indicating a default event (i.e., the obligor is categorized as *bad* at the end of the observation period). Finally, some necessary data cleansing is performed on the available data sheets, to remove entries with missing values. This way, we eventually obtain a dataset comprising 8244 obligors, which mainly include Greek SMEs (total assets worth less than 50 Million Euro), as well as some large corporations.

To develop our model, we split the so-obtained dataset into three parts: An *in-sample* dataset, comprising data pertaining to the 70% of the examined companies, obtained over the observation period 2006-2011; an *out-of-sample* dataset, comprising the data pertaining to the rest 30% of the companies for the period 2006-2011; and an *out-of-time* dataset that comprises all the data pertaining to the observation period of year 2012. A summary of the aforementioned split of our dataset is provided in Table 17. In Table 18, we provide a brief summary of the breakout of the used data, showing the numbers of the available samples that pertain to SMEs and large corporations, respectively.

Table 17: Dataset split into in-sample, out-of-sample, and out-of-time sets.

Dataset Split	Good Obligors	Bad Obligors	Default Rate (%)	Total
in-sample	5513	328	5.62	5841
out-of-sample	1652	100	5.7	1752
out-of-time	536	115	17.7	651
Total	7701	543	6.58	8244

Table 18: Distribution of the used data by asset size.

Assets (in Million Euro)	Frequency
5	1837
50	5028
100	707
200	338

500	190
>500	144
Total	8244

System development, calibration, and analysis is performed using our in-sample dataset. The out-of-sample and out-of-time datasets are in turn used to perform system evaluation under two different scenarios: Evaluation of the generalization capacity of our system across companies, and evaluation of the generalization capacity of our system over time. To allow for reliable estimation of the hyperparameters of our system (i.e., of the number of hidden states, I, of the postulated SHMMs, and the number of components, J, of the entailed Student's-t mixture models), we further split our in-sample dataset into a training set and a validation set: the training sample is used to train the postulated SHMMs pertaining to each financial ratio and each of the two obligor characterizations (*good* or *bad*), while the validation set is used for model selection, i.e. optimal determination of the model hyperparameters (model size).

Data Processing and Transformation. For each examined corporation, we elect to model a set of well-known and broadly used financial ratio time-series, extracted by exploiting their available balance sheets and income statements. Specifically, the used set of financial ratios comprises the following indices:

- A set of broadly used financial ratios reflecting *liquidity*, including: (i) current ratio (X1); (ii) immediate cash ratio (X2); (iii) working capital (X3); (iv) total employed capital (X4).

- A set of financial ratios that reflect *profitability*, including: (i) return on equity (X5); (ii) return on total employed capital (X6); (iii) gross profit margin (X7); (iv) operating profit margin (X8); (v) net profit margin (X9).
- A set of financial ratios that reflect *capital structure,* including: (i) fixed assets coverage ratio (X10); (ii) leverage ratio (X11); (iii) interest coverage (X12); (iv) equity over employed capital (X13).
- A set of financial ratios that reflect *activity*, including: (i) receivables turnover ratio (X14); (ii) trade creditors to purchases ratio (X15); (iii) inventories turnover ratio (X16); (iv) employed capital turnover ratio (X17); (v) equity turnover ratio (X18).

In addition to these standard financial ratios, we have also experimented with various transformations of these ratios, with the aim to obtain more representative financial times-series to train our credit rating models with. Specifically, for this purpose, we subsequently followed three distinct procedures:
(i) We applied a series of simple transformations on the original time-series, including *square*, *cube power*, *log*, *sin*, *1/(1+x),* and *inverse*.
(ii) Subsequently, we computed the year-over-year percentage changes of the considered time-series.
(iii) Finally, in an effort to obtain more robust input variables to train our models upon, we generated 50,000 random (derivative) financial ratios based on the available datasets. For this purpose, we followed an iterative procedure that consists in randomly selecting 4 original items of the original balance sheets and income statements, say a, b, c, and d, and computing a derivative ratio of the form $(a \pm b)/(c \pm d)$

.

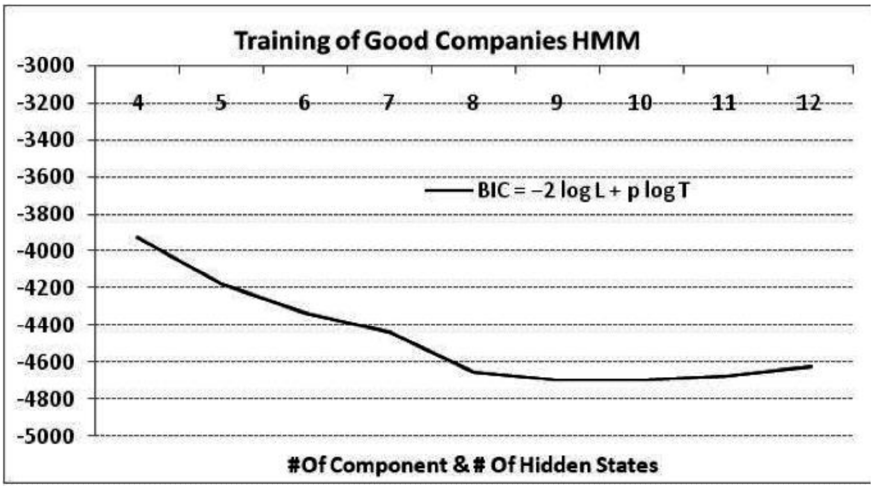
This process led to a set of almost 2,000 predictor variables (distinct time-series) as potential candidates for our modeling procedures. The so-obtained set of time-series was narrowed down in three consecutive stages: On the first stage, we kept the 200 time-series exhibiting the highest in-sample correlation with the modeled (binary) dependent variable, i.e. the categorization of obligors as *good* or *bad* at the end of the observation period. On the second stage, we omitted those of the aforementioned 200 time-series that bear no economic meaning/intuition. Finally, on the third stage, we narrowed down the selected variables (derivative financial ratios) by setting a threshold of at least +/- 10% correlation with the *default* flag variable. This way, we eventually retained 8 new *derivative* financial ratio time-series that we use to perform model training in the context of our system, additional to the previously mentioned, commonly used ones. Note that we have transformed the above-mentioned ratio values into the [0, 1] interval, using a simple linear transformation [172].

These newly-obtained financial ratio time-series are namely the following: (i) (Operating profit-Interest Expenses) / Sales (X19); (ii) (Short term liabilities + Cost of Sales) / Sales (X20); (iii) (Long term liabilities - Gross profit) / Total Assets (X21); (iv) Bank Loans/ Gross profit (X22); (v) (Gross profit + Equity)/Total Liabilities (X23); (vi) Current Assets/Sales (X24); (vii) Borrowed funds/ Turnover (X25); and (viii) the *1/(1+x)* transform of the interest coverage ratio (X26). Note that the first seven derivative financial ratio time-series mentioned above essentially reflect *capital structure,* thus bearing a clear financial intuition/relevance.

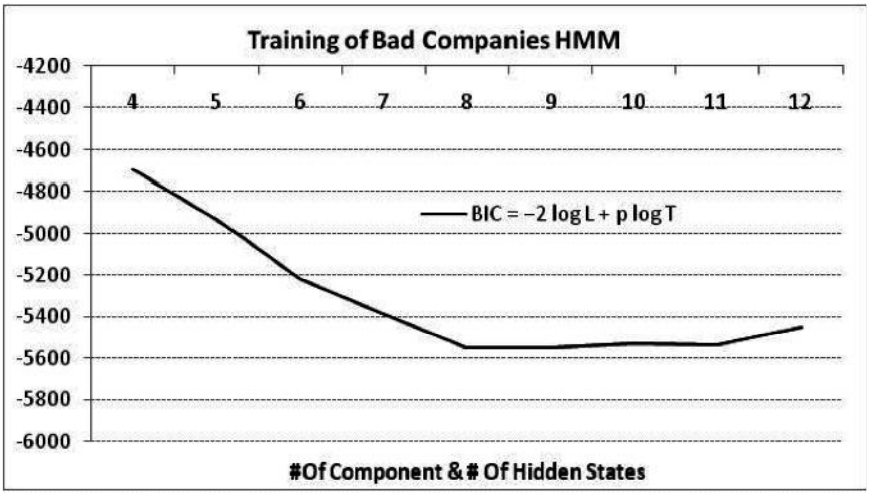
5.4.2.2 HMM Model Training

As previously discussed, for every examined company and for every modeled financial ratio, we build a time-series comprising values recorded over five consecutive years. This dataset is subsequently used to perform model training. As we have also discussed, each sequence is categorized as good or bad, depending on the corresponding obligor performances. To effectively model these data in the context of the proposed system, we postulate two distinct SHMMs for each financial ratio, one pertaining to obligors categorized as good, and one pertaining to obligors categorized as bad. Our modeling selection allows for capturing salient temporal patterns and dynamics in our modeled time-series, in an effort to detect shifts in the state of the economy and their correlations with changes in the behaviour patterns of the examined companies. As we have discussed, we split our in-sample dataset into one training set and one validation set. The training set comprises 200 sequences for each financial ratio, and for each category of obligors (good or bad), either large corporations or SMEs. Parameter initialization (before model training using the EM algorithm, described in Section 3) was performed using the segmental K-means algorithm described in [4]; the degrees of freedom ν of the Student's-t distributions are initialized at $\nu = 1$.

Turning to model (size) selection, in each case our selections are made between models comprising 2-6 mixture component Student's-*t* distributions, and 2-6 hidden states. To perform model selection, at first we utilize the popular Bayesian information criterion (BIC) [13]. BIC is widely used for selecting proper model size in the HMM literature, by appropriately penalizing the obtained log-likelihood of the trained model with a penalty term the accounts for the number of postulated parameters (i.e., model size), to prevent overfitting. In the context of our system, we utilize BIC so as to retain the 10 highest-ranked possible model configurations (out of a set of 25 initial alternatives). To alleviate the effect of random initialization (of the segmental K-means algorithm) on the obtained results, we repeat training of each considered model multiple (namely, 20) times, and retain the random restart that yielded the best BIC value. In Fig. 5.4, we show how BIC values change (on average over the modeled assets) with model size in our experiments, both in cases of models pertaining to *good* obligors, and in cases pertaining to *bad* obligors.



(a)



(b)

Figure 5.4: BIC values as a function of model size.

Further, for each of the retained model configurations (sizes), we use the forward algorithm to compute the likelihood of each corresponding time-series in our validation set. We perform this procedure with respect to both the corresponding postulated SHMMs pertaining to obligors categorized as *good,* and the corresponding postulated SHMMs pertaining to obligors categorized as *bad,* and compute the *log-ratio* of the two *likelihoods.* Finally, we rank the postulated alternative SHMM configurations on the basis of the correlation of these obtained (*good* to *bad*) *likelihood log-ratios* with the actual ratings of the considered obligors: the higher the value of these log-ratios, the more correlated they are with obligors actually rated as *good,* and the less correlated they are with obligors actually rated as *bad.*

On the basis of this procedure, for each financial ratio we retain the *good/bad* obligor SHMM pair configuration that yields the best ranking among the considered alternatives. In Table 19, we depict the average obtained SHMM size (over all the modeled financial ratios) separately for the models fitted to data from *good* obligors, and for the models fitted to data from *bad* obligors. As we observe, to sufficiently capture the underlying temporal patterns, companies rated as *good* require larger models than companies rated as *bad.* This is a rather intuitive result, since companies rated as *good* are expected to exhibit more heterogeneous patterns than companies eventually defaulting on their debt.

Table 19: Average optimal size of trained SHMMs by obligor category.

Category	*Good*	*Bad*
# Mixture components	4.00	4.06
# States	5.06	4.09

5.4.2.3 Model Aggregation

After obtaining the component SHMMs of our system, we apply a sample scoring procedure for the entire in-sample dataset. Specifically, for every company in our in-sample dataset, we produce a 26-dimensional vector containing the *likelihood log-ratio scores* pertaining to the two trained SHMMs (*good/bad*) for each modeled financial ratio (for this purpose, the forward algorithm is used as described in Section 3.3). As previously discussed, the *likelihood log-ratio scores* of a modeled company essentially encode how likely our trained models consider the company to end up with a *good* rating at the end of the observation period. Apparently, as a result of our modeling choices, each pair of postulated SHMMs (modeling a different financial ratio) generates a different *likelihood log-ratio score.* Hence, it is necessary that we come up with an optimal way of combining these scores so as to derive a final predictive score from our model.

For this purpose, we postulate a simple linear score combination model driven by the *likelihood log-ratio scores* generated as described previously. To train this model, we use a genetic algorithm [160]that aims to maximize the overall score correlation with the dependent variable (good/bad obligor flag) over the modeled in-sample population. Our selection of the aforementioned genetic algorithm as the optimization method of choice is motivated from its simple black-box nature, and its attractive properties it terms of the obtained rates of convergence to the global optimum of the solved complex optimization problem. We experiment with various mutation rates and numbers of generations, in order to select the optimal genetic algorithm configuration. For completeness sake, in Table 20 we provide the final linear model parameter (weight) values estimated through the used genetic algorithm.

5.4.2.4 System Calibration

Eventually, we utilize the credit score values generation capabilities of our system to obtain a *default probability* prediction mechanism. For this purpose, we apply a credit rating system *calibration* process. Calibration of a credit rating system is a mapping process under which each possible generated score value is allocated an associated probability of default. To perform calibration of our system on the in-sample population, we divide the set of obtained *likelihood log-ratio* scores, generated in the previous stage, into ranges. Each range is associated with a probability of default. Computation of ranges is performed in such a way that ensures maximum intra-rate homogeneity of the obtained probabilities of default, and maximum inter-range heterogeneity. To achieve this, we use a well known discretization algorithm, namely MDLP; our adopted algorithm follows the minimum description length (MDL) estimation principle [173], which optimizes continuous variable ranges based on a class entropy criterion. Finally, we correct for monotonicity (if needed) by fitting an exponential function.

Table 20: Financial ratio weights obtained by application of the used genetic algorithm.

Financial Ratio	Weight
X1	3.8
X2	6.2
X3	9.2
X4	3.4
X5	1.4
X6	6.3
X7	0
X8	2.7
X9	6.1
X10	1.6
X11	1.7
X12	2.4
X13	3.6
X14	1.6
X15	6.5
X16	0
X17	0
X18	1.9
X19	4.9
X20	3.3
X21	3.1
X22	6.6
X23	3.1

X24	5.6
X25	3.1
X26	12.2

5.5 Experimental Evaluation

Here, we report the performance results obtained from the experimental evaluation of our method, both in terms of out-of-sample performance, and in terms of out-of-time performance. To obtain some comparative results, apart from our method we also evaluate a set of established benchmark models in the field of corporate credit rating, namely CHAID, LDA, logistic regression, SVMs, RFs, and FNNs. Since the considered benchmark approaches are not capable of modeling time-series data, we opt to retain from (the corresponding time-series of) each financial ratio only those of the five constituent observed variables that do not exhibit strong inter-correlations. Specifically, to perform this procedure, we first compute the Pearson correlation matrix of the available 5-year data (of each financial ratio). On this basis, we exclude the time point variables exhibiting more than 60% absolute correlation with (some) other time point variables, to avoid multicollinearity. Eventually, for each pair of correlated variables, we retain the one that exhibits higher correlation with the dependent variable and drop the other one.

We implemented our method in Microsoft Excel Visual Basic (VBA). We also used the SolveXL add-in of Microsoft Excel to perform genetic algorithm-based optimization. We implemented the MDLP algorithm based on the Discretization package of R.

The remainder of this section is organized as follows: In Section 5.1, we describe the details of our implementation of the considered benchmark approaches (evaluated in parallel to our method). In Section 5.2, we provide an analytical account of our experimental results, and discuss how performance of our method compares to the competition.

5.5.1 Benchmark Models Implementation

5.5.1.1 CHAID

CHAID is well-established algorithm for building decision trees [139]. Similar to other decision trees algorithms, CHAID allows for simplicity and intuitive visualizations of the obtained results. In addition, the non-parametric nature of CHAID allows for increased flexibility compared to other regression models. Nevertheless, these simplicity advantages come at the cost of significant overfitting proneness due to the entailed discretization of the observed time-series. In our experiments, variable discretization is performed by utilizing 10-bin histograms. We implemented CHAID using the XLSTAT package for VBA.

5.5.1.2 LDA

LDA is broadly used for credit scoring. For instance, the popular Z-Score algorithm of [133]is based on LDA. In essence, LDA is used to build binary classification models, predicting whether an examined company will go bankrupt or not. LDA is based on two main assumptions: (i) that the modeled independent variables are normally distributed; and (ii) that the two groups of modeled obligors (*good* and *bad*) exhibit homoscedasticity. As we previously discussed though, these assumptions are hardly plausible in real-world financial time-series. We implemented this approach in R, using the MASS R package.

5.5.1.3 Logistic regression

Logistic regression is very often used by financial institutions for building credit scoring models due to its parsimonious structure. Similar to LDA, it is used to estimate the non-linear relationship between the modeled continuous independent variables and a categorical/binary dependent variable (in our case, *good* or *bad* obligors). In our implementation, model training is performed using maximum-likelihood estimation. To perform optimization in the context of the M-step of the algorithm, we resort to the Newton-Raphson iterative optimization method. We implemented logistic regression in VBA, using the XLSTAT add-in.

5.5.1.4 SVMs

SVMs are one of the most popular types of non-linear, large-margin binary classifiers, estimating a separating hyperplane that achieves maximum separability between the data of the modeled two classes [123]. In our study, we evaluate *soft-margin* SVM classifiers using linear, radial basis function (RBF), polynomial, and sigmoid kernels, and retain the model configuration yielding optimal performance. For the latter purpose, we exploit the available validation set. Similarly, to select the hyperparameters of the evaluated kernels, as well as the cost hyperparameter of the SVM (related to the adopted soft margin), we resort to cross-validation; the candidate values of these hyperparameters are selected based on a *grid-search algorithm* [123]. We implemented this model in R using the e1071 package; grid-search is a functionality included in the e1071 package (*Tune* routine). The employed cross-validation procedure determines the optimal SVM structure to comprise a linear kernel function with cost hyperparameter equal to 150.

5.5.1.5 RFs

RFs have recently received considerable attention in various financial research fields [124]. RFs are supervised statistical machine learning methods that combine bootstrap aggregation and random subspace selection to generate or grow trees that all together define a forest. In more detail, RFs combine many binary regression decision trees that are selected by bootstrapping samples of the modeled explanatory variables and the corresponding classifier variables. Final prediction is made by averaging the predictions from all the individual trees in cases of regression problems, or using majority voting in cases of classification problems. The final set of random forest variables is selected using a variable importance index, which reflects the ‘‘importance’’ of a variable based on its contribution to classification accuracy. This is estimated by looking at how much prediction error increases when omitting a considered variable. Our implementation of RFs was based on the randomForest package of R. To perform optimal selection of the maximum number of trees in the forest, we perform cross-validation using the available validation set; we select among specifications comprising 20, 50, 100, 200, 500, 600, 700, 800, 900, and 1000 trees. This procedure yields a forest comprising 600 trees. The maximum number of selected variables for each tree is set equal to the 1/3 of the available financial ratios.

5.5.1.6 FNNs

Typically, credit rating systems employ multilayer perceptron (MLP) FNNs comprising the following layers: the *input* layer, where the explanatory variables are presented to the network, *one or more* hidden layers comprising sigmoid transfer functions, and an output layer where the predicted values are generated [174]. To perform model training, we use back-propagation [174]. We perform cross-validation to select the number of hidden layers and their component hidden neurons, exploiting the available validation set. This procedure selects an MLP with 1 hidden layer and 21 hidden neurons.

Training algorithm hyperparameters, including learning rates and momentum values, are also selected by means of cross-validation. Early stopping is employed to avoid overfitting. Our implementation of MLPs was based on the NeuroSolutions toolbox for VBA.

5.5.2 Comparative Results

5.5.2.1 Discriminatory Power Results

High discriminatory power is a key requirement for rating systems, and the main evaluation criterion for selecting between alternative rating approaches. To quantitatively measure the performance of a scoring model, researchers and practitioners typically use statistical measures of performance such as the area under the receiver operating characteristic (ROC) curve, the GINI coefficient (accuracy ratio), the Kolmogorov-Smirnoff (K-S) statistic, the Bayesian error rate, Kendall's τ and Somer's D. In this work, we assess the discriminatory power of the evaluated rating systems using the GINI metric, K-S metric, and obtained Bayesian error rates [132]. Area under the ROC curve is not used, as it is directly connected with GINI, and essentially captures the same performance characteristics. Similarly, Kendall's τ and Somer's D usually provide similar insights with the aforementioned statistical measures, and, therefore, we decide to omit them from our analyses [132].

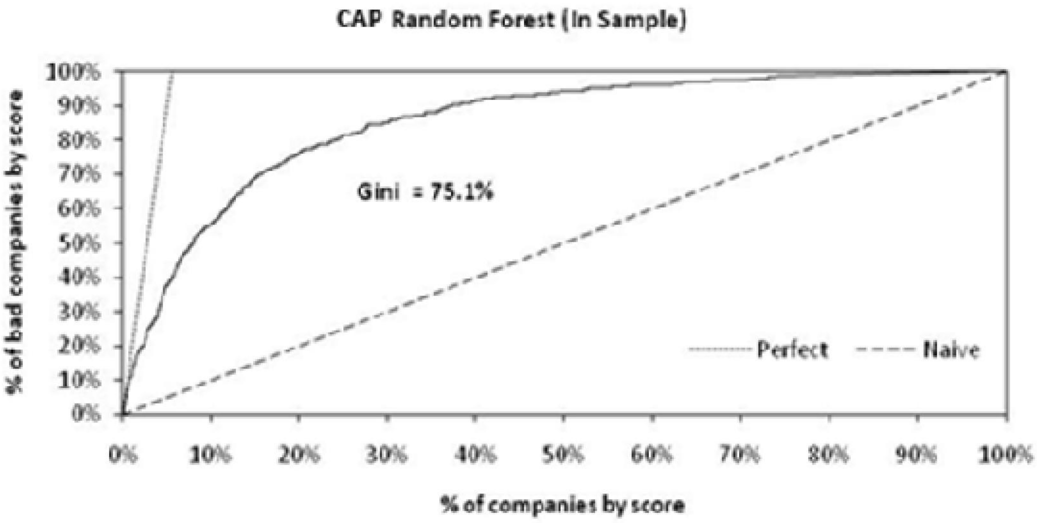
In Table 21, we depict the results obtained from the evaluated models. It is evident that the proposed SHMM-based rating system exhibits higher discriminatory power compared to all the considered competitors. More significantly, the obtained performance is more stable and more consistent across all test samples, resulting in lower performance standard deviation. This is an important merit of our approach, since achieving high average performance is as significant for a rating system as it is for it to achieve low performance variance, and thus, higher consistency and better performance guarantees.

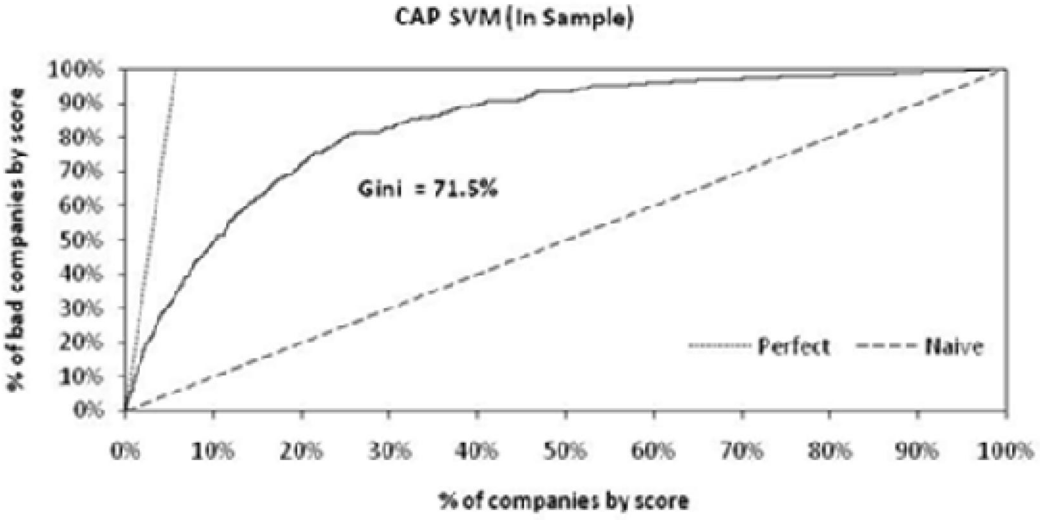
Another interesting finding stemming from our results is that CHAID performs very poorly in the cases of the out-of-sample and out-of-time datasets. To our perception, this finding is most likely due to overfitting. On the other hand, we observe that FNNs perform slightly better than logistic regression and other machine learning techniques. Note though that the performance superiority of FNNs is not significant enough to counterbalance the advantages of other machine learning approaches, such as LDA and RFs, which offer much better computational complexity, while RFs have also the major advantage of allowing for yielding intuitive visualizations of the results of the inference algorithm.

Regarding the obtained Bayesian error rate, we observe that our results confirm the stability of our approach, since the values of this statistic are similar in all the considered scenarios (in-sample, out-of-sample, and out-of-time), and the classification errors are significantly lower than the considered benchmark models. Finally, we underline that the obtained GINI performance of our model is equal to or greater than 80% in all cases; according to industry benchmarks, SME credit rating systems yielding a GINI index exceeding 80% are considered to possess significantly high (industry-level) discriminatory power. Hence, our approach possesses the significant merit of yielding industry-level predictive performance, which increases its potential attractiveness to real-world financial institutions. A graphical illustration of the evolution of the obtained GINI values is provided in Fig. 5.5.

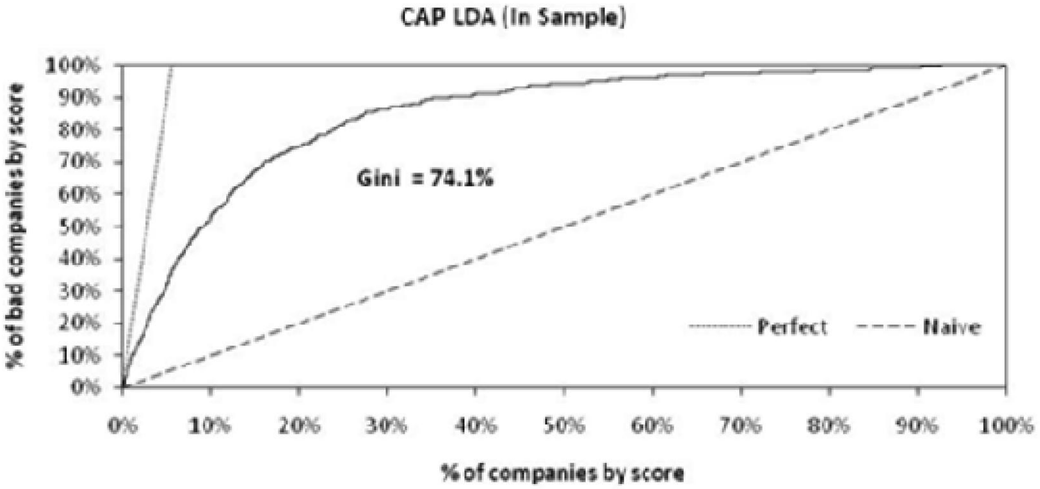
Table 21: Discriminatory Power Results of the Evaluated Algorithms.

Test	Logistic regression	Neural network	CHAID	LDA	SVM	RandomForests	HMM
GINI							
In-Sample	75.1%	75.3%	70.6%	74.1%	71.5%	75.1%	**79.9%**
Out-of-Sample	73.8%	74.5%	55.2%	71.2%	65.4%	71.5%	**80.9%**
Out-of-time	78.2%	79,5%	64.0%	75.4%	76.0%	77.8%	**84.3%**
K-S							
In-Sample	61.8%	62.1%	51.7%	61.2%	59.2%	60.2%	**67.4%**
Out-of-Sample	60.6%	63.4%	42.3%	57.5%	54.5%	59.2%	**68.9%**
Out-of -time	65.8%	66.7%	52.5%	63.4%	63.1%	63.7%	**73.3%**
Bayesian Error Rate							
In-Sample	24.0%	19.0%	38.4%	25.0%	24.7%	22.4%	**13.9%**
Out-of-Sample	17.5%	16.4%	37.4%	18.4%	19.7%	18.1%	**12.8%**
Out-of-time	16.6%	15.1%	32.0%	17.8%	20.5%	16.9%	**12.1%**

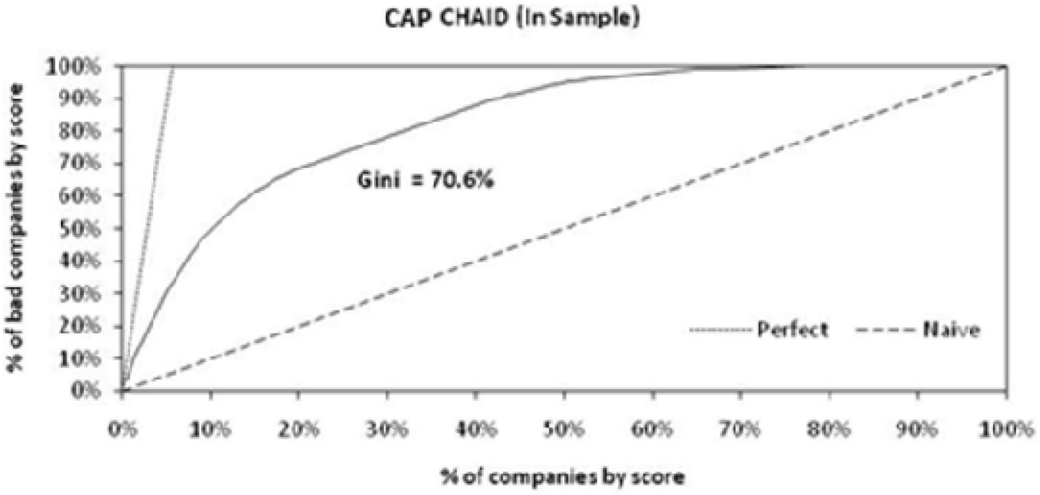




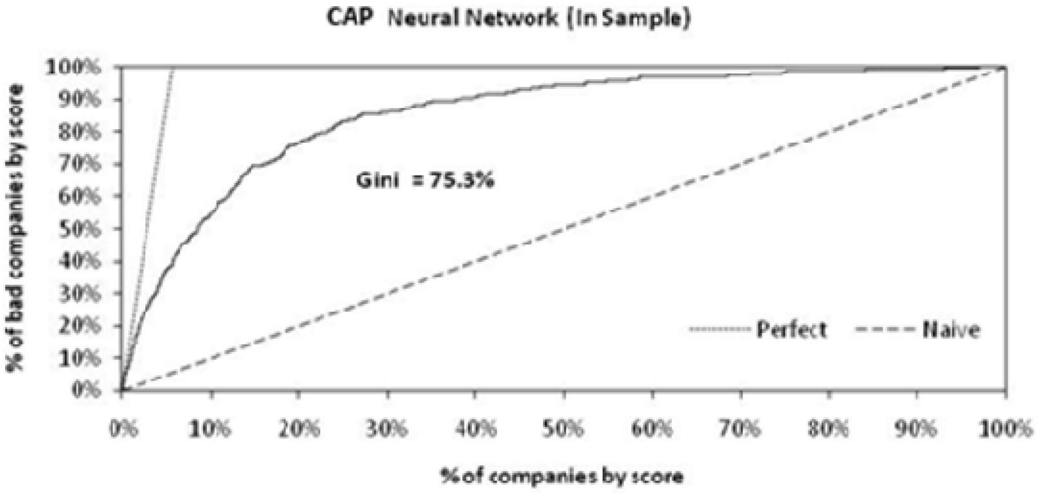
CAP SVM (In Sample)

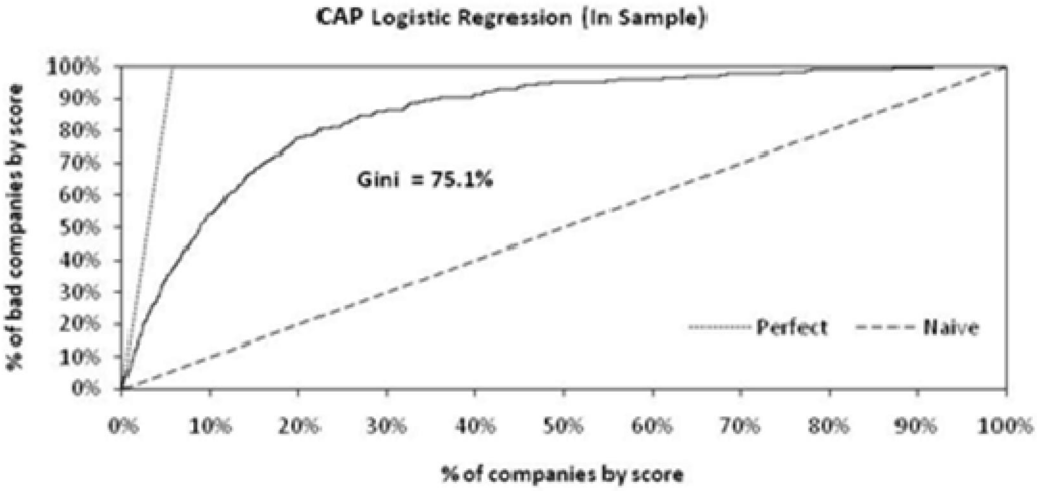


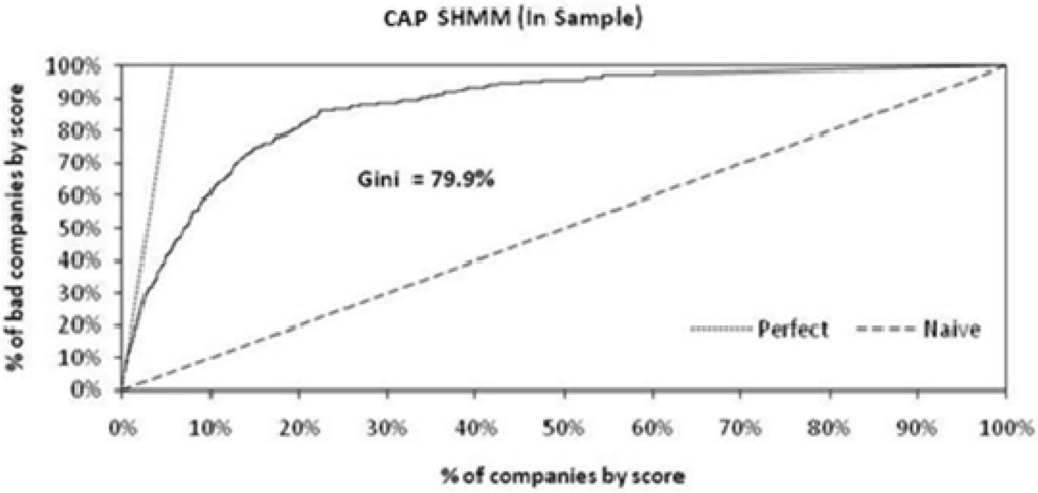
CAP LDA (In Sample)

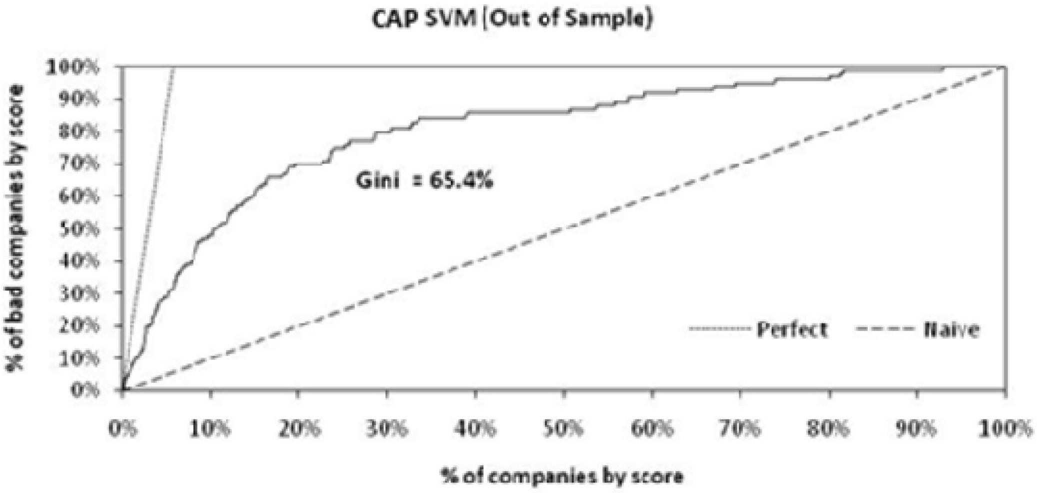


CAP CHAID (In Sample)

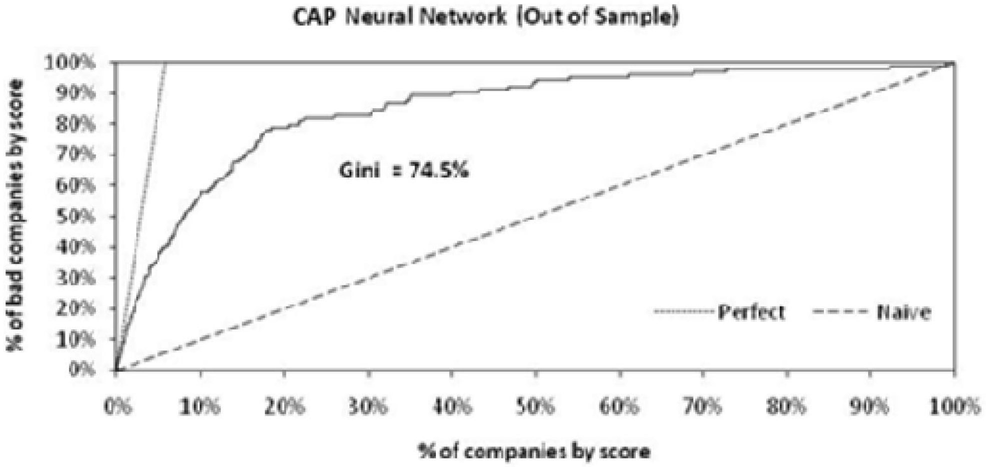

CAP Neural Network (In Sample)
Gini = 75.3%
Perfect
Naive
% of bad companies by score
% of companies by score
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%


CAP Logistic Regression (In Sample)
Gini = 75.1%
Perfect
Naive
% of bad companies by score
% of companies by score
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

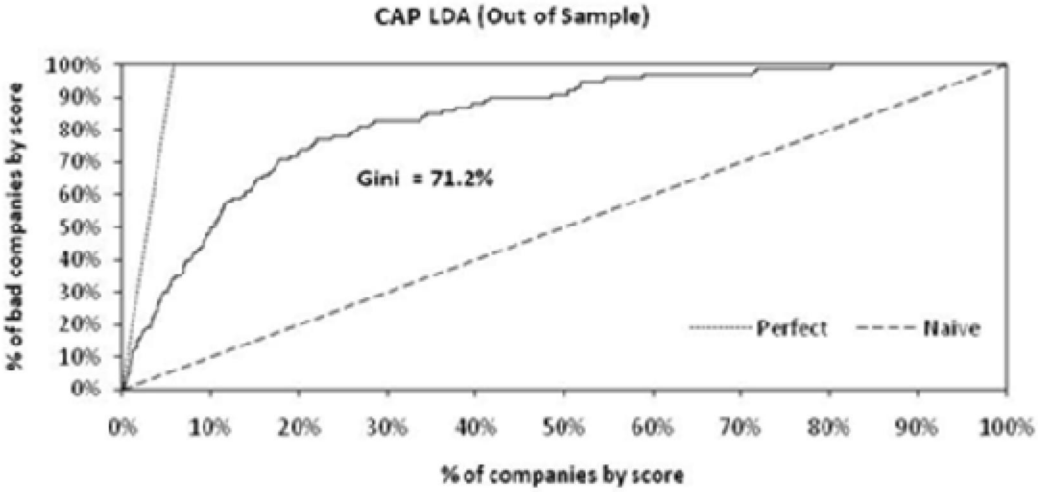

CAP SHMM (In Sample)
Gini = 79.9%
Perfect
Naive
% of bad companies by score
% of companies by score
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%



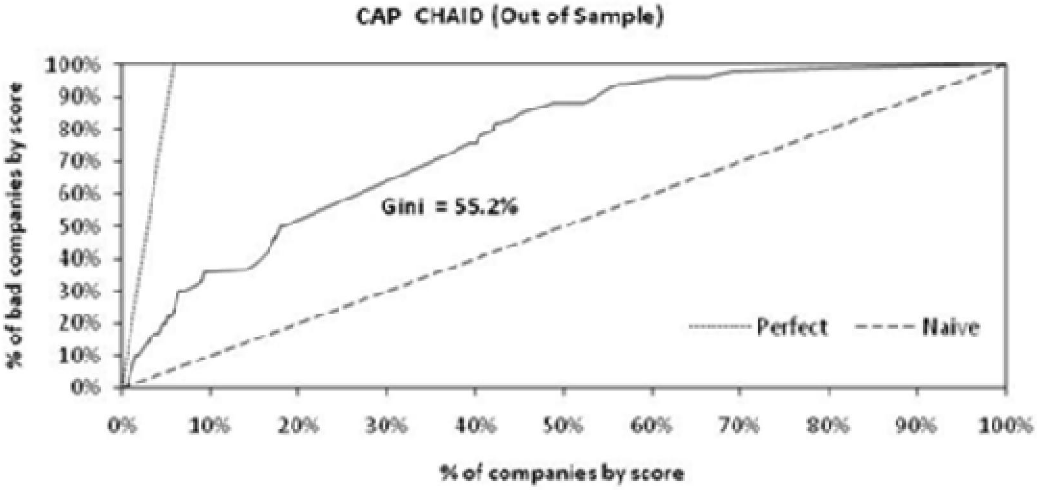
CAP SVM (Out of Sample)

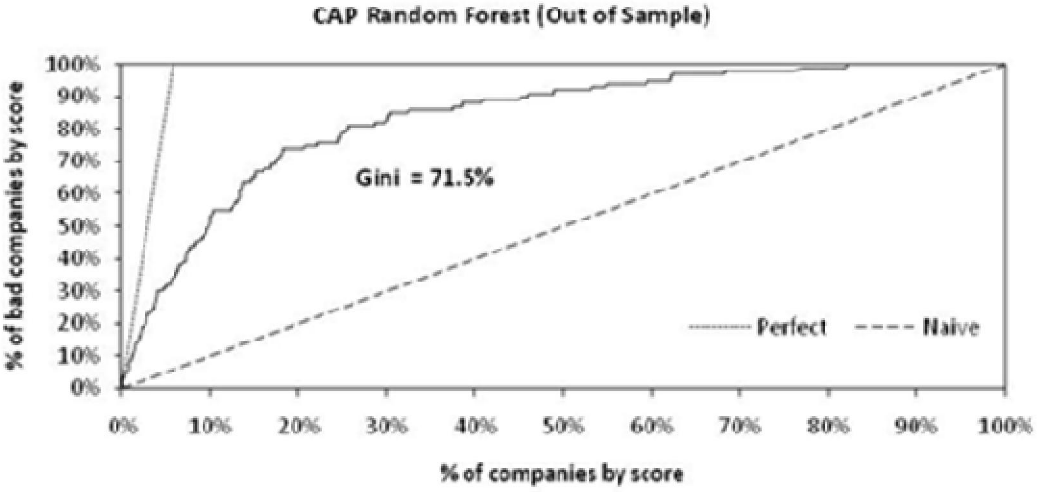


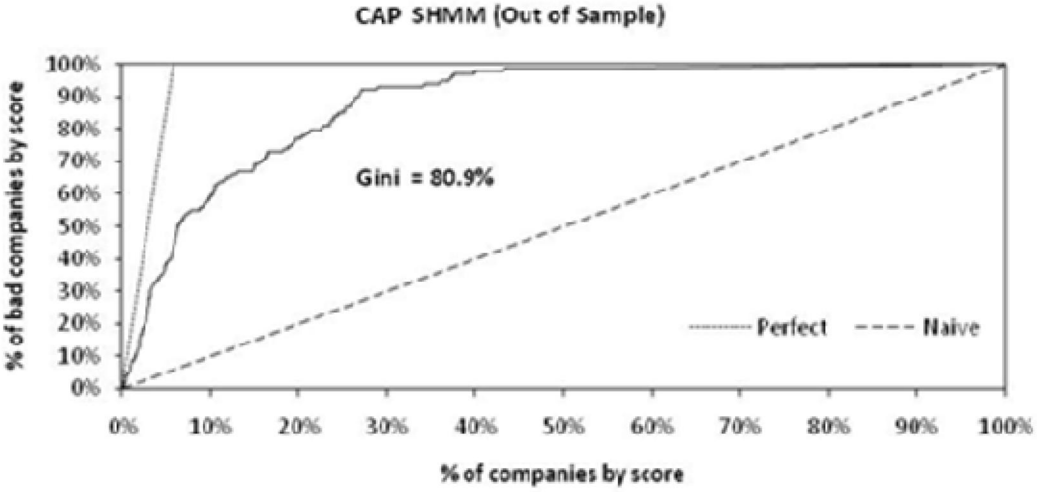
CAP Neural Network (Out of Sample)

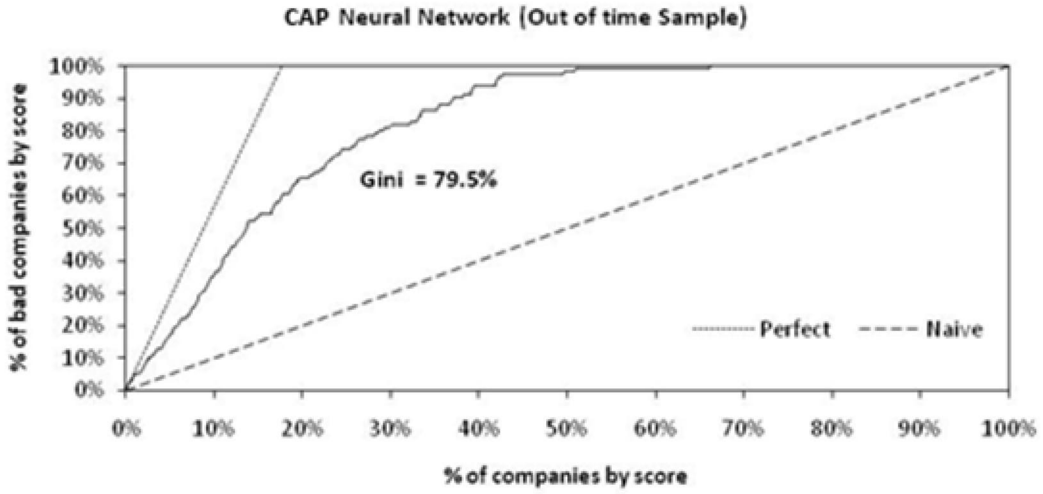


CAP LDA (Out of Sample)

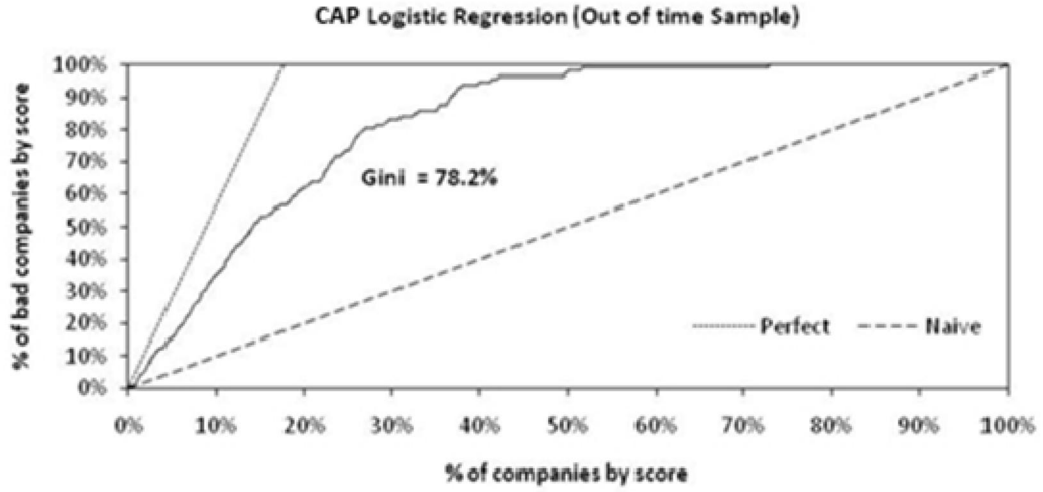

CAP CHAID (Out of Sample)
% of bad companies by score
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Gini = 55.2%
Perfect
Naive
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
% of companies by score


CAP Random Forest (Out of Sample)
% of bad companies by score
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Gini = 71.5%
Perfect
Naive
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
% of companies by score

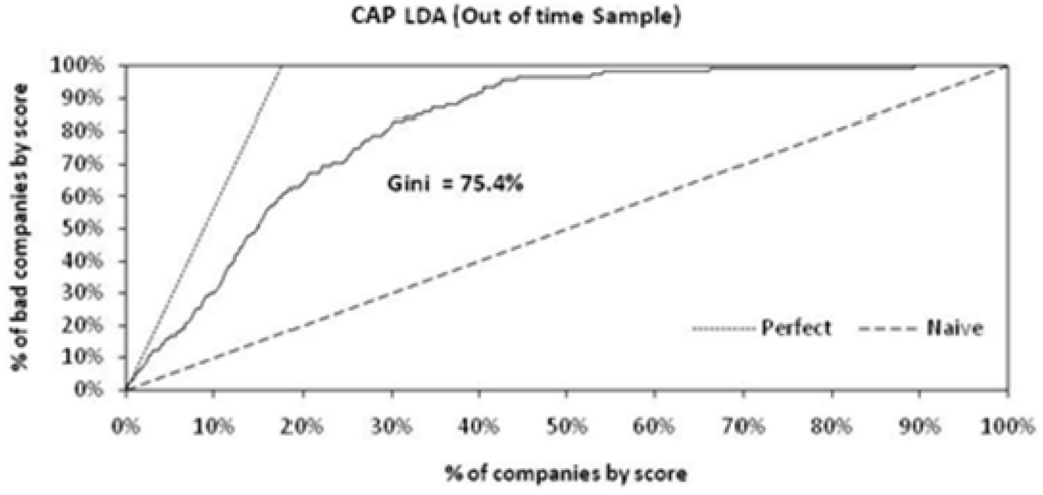

CAP SHMM (Out of Sample)
% of bad companies by score
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Gini = 80.9%
Perfect
Naive
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
% of companies by score



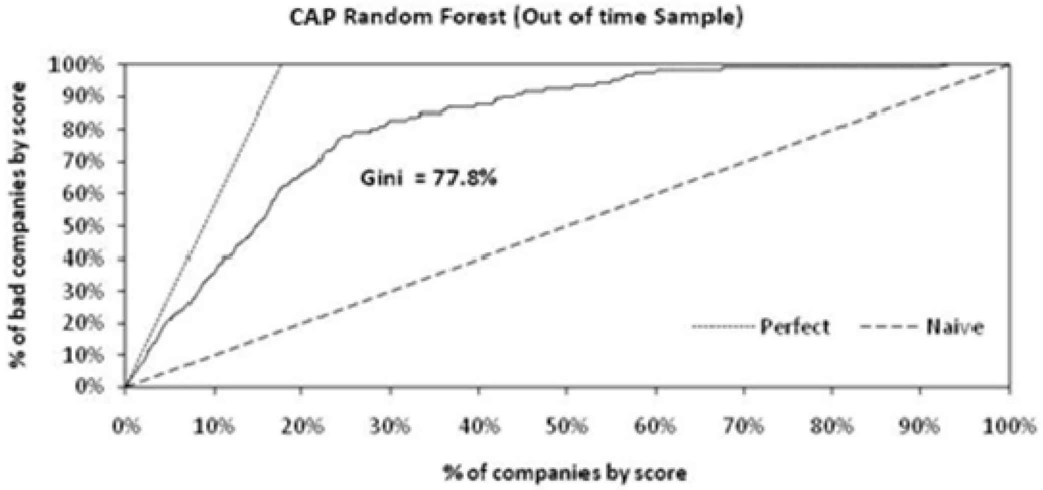
CAP Neural Network (Out of time Sample)



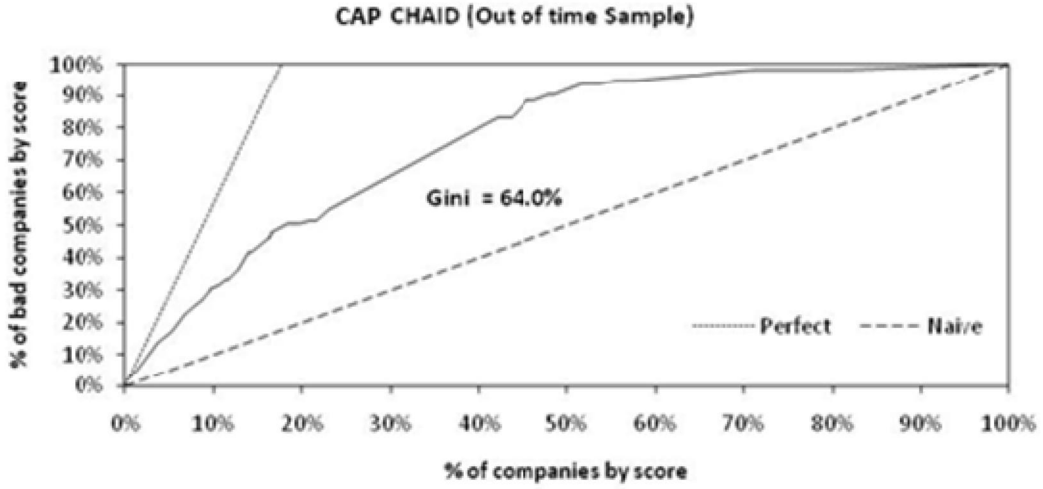
CAP Logistic Regression (Out of time Sample)



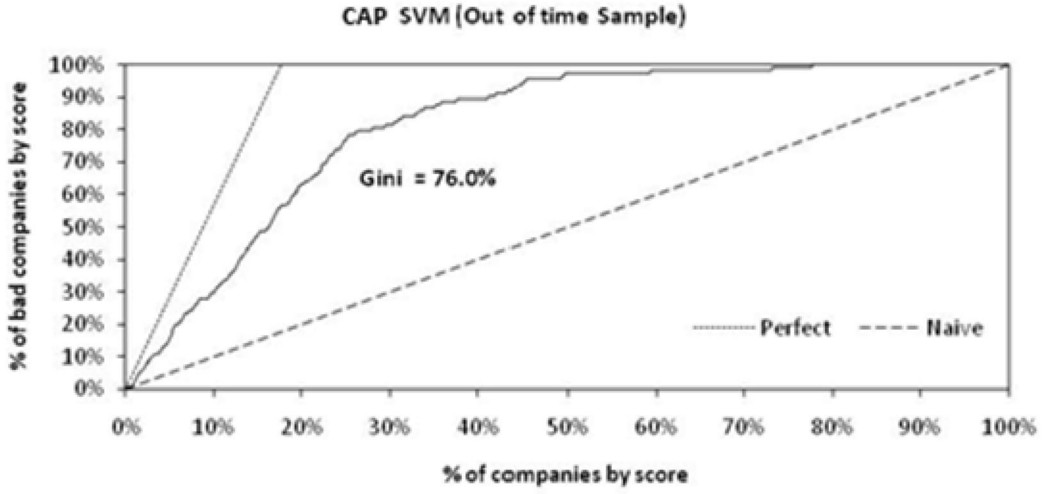
CAP LDA (Out of time Sample)



CAP Random Forest (Out of time Sample)



CAP CHAID (Out of time Sample)



CAP SVM (Out of time Sample)

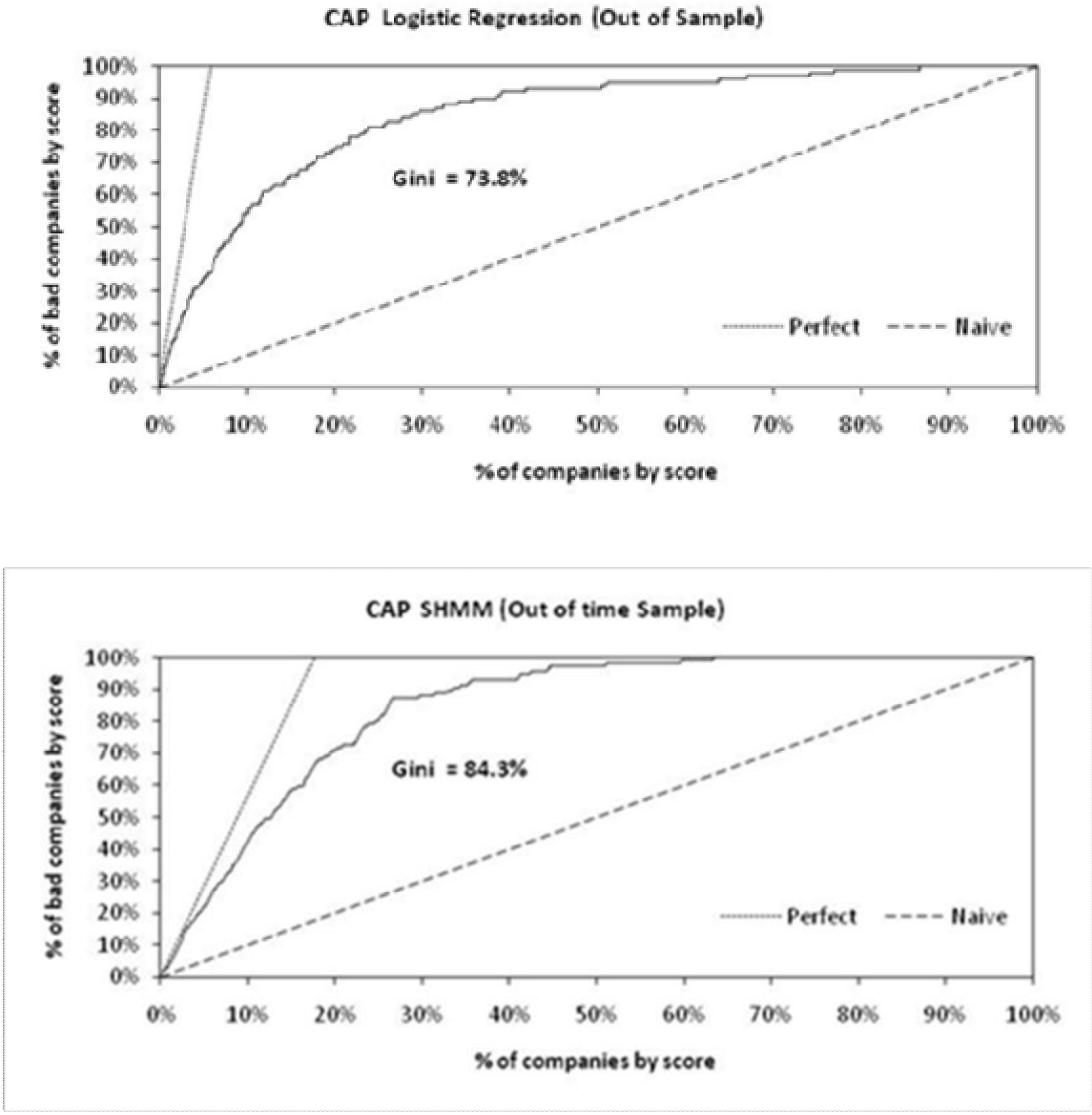


Figure 5.5: Analysis of obtained GINI performance values.

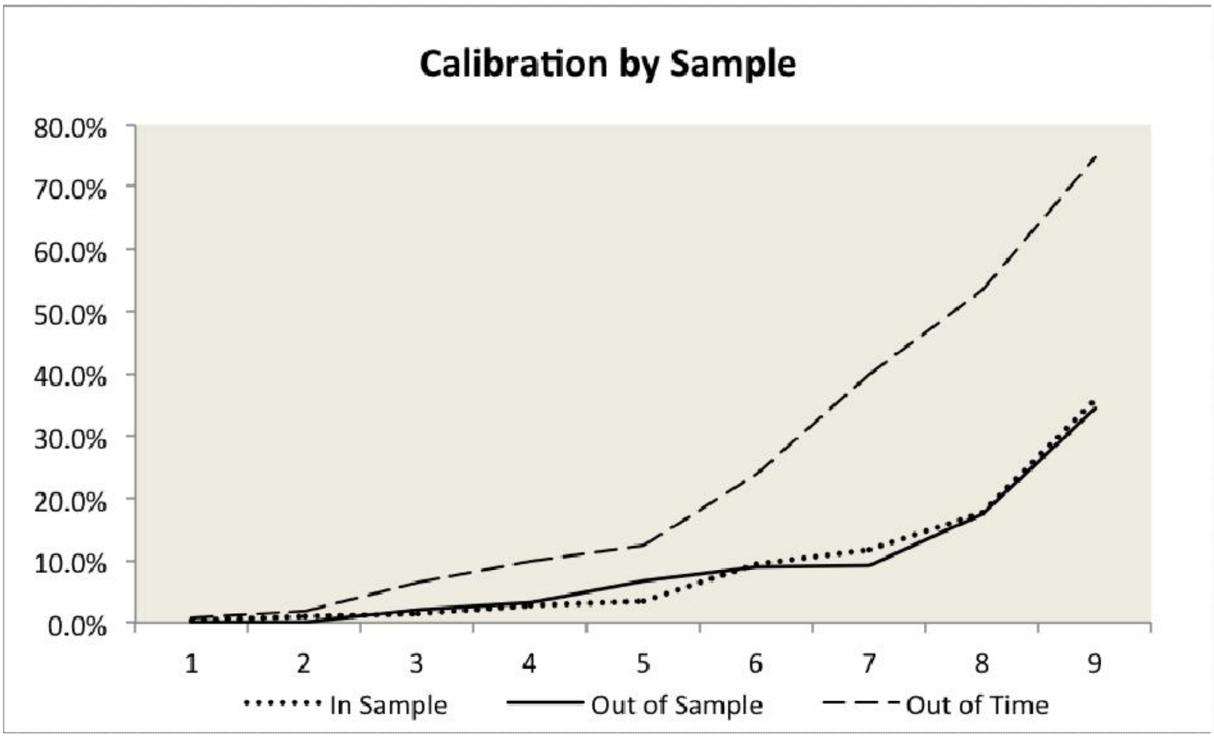


Figure 5.6: Calibration Results.

5.5.2.2 Calibration Results

Finally, we elaborate on the results obtained from the system calibration procedure described in Section 4.2.4. This procedure yields 9 rating grades. The default rates obtained by the calibrated 9-grade rating system are depicted in Fig. 5.6 (solid black line). In the same figure, we also show the results obtained from applying calibration to the out-of-sample and out-of-time datasets, using the rate ranges determined on the in-sample population. Looking at the out-of-sample results, we observe a rather stable performance in the estimation of the actual default rate that corresponds to the out-of-sample population. This is also verified by a performing a chi-square test (Table 22) to compare the in- and out-of-sample calibrated populations. Therefore, we deduce that our SHMM-based prediction system does not exhibit statistically significant performance differences between the in-sample and out-of-sample datasets.

Further, we perform a similar analysis regarding the out-of-time samples. In this case, we observe quite different a result: Indeed, we observe a deviation of obligor's behaviour (implied default rate) equal to 5.6% w.r.t. the in-sample dataset, and equal to 17.7% w.r.t. the out-of-sample dataset. We would like to underline that this is not an unexpected system behaviour: credit rating systems typically need recalibration in their rating scale when dealing with out-of-time datasets, in order to allow for capturing significant changes in the business environment that cannot be otherwise predicted using the modeled financial ratio time-series. To resolve this issue, one could consider introducing into the fitted models some additional macroeconomic variable as a covariate (e.g., GDP, unemployment rate).

Table 22: Calibration Results: Chi-squared test outcomes.

	out-of-sample	out-of-time
χ^2value (8 DoF)	8.8	288
p-value	>0.10	<0.0001
Null Hypothesis	Accept	Reject

Chapter 6 Conclusions and ideas for further research

In this final chapter we review our main results and discuss possible extensions to our work. The common theme of all chapters in this thesis is the application of HMMs in finance. After investigating the evolution of HMM and their theory in the last decades and tracking down the most significant results in the field of finance we extended theoretically the framework of HMMs by proposing a variable order with dependence jumps variation. We thorough investigated its forecasting ability in a broad group of financial time series and benchmarked our result with a wide group of state of the art models. Finally we designed a novel approach for building a corporate credit rating system using Student-t Hidden Markov Models that exhibits increased efficiency and stability.

6.1.1 VDJ-HMM variation

In chapter 4, we focused on the problem of modeling sequential data the temporal dynamics of which may switch between different patterns over time. To address this problem, we introduced a hierarchical model comprising two hidden chains of temporal dependencies: on the *first layer*, our model comprises a *chain* of *latent observation-emitting states,* the *dependencies* between which may *change over time*; on the *second layer,* our model utilizes a *latent first-order Markov chain* modeling the *evolution* of temporal dynamics pertaining to the first-layer latent process. To allow for tractable training and inference procedures, our model considers *temporal dependencies* taking the form of *variable order dependence jumps*, the order of which is *inferred* from the data as part of the model inference procedure. We devised efficient model training and inference algorithms under the maximum-likelihood paradigm.

To evaluate the capacity of our method in effectively modeling non-homogeneous observed sequential data, where the patterns of temporal dependencies may change over time, we considered a number of computational finance applications. Specifically, we considered both *volatility forecasting* applications as well as *value prediction* applications dealing with financial return series for sets of considered assets. As we discussed, this setting allows for an objective evaluation of whether our approach does actually achieve its goals, since empirical evidence has shown that financial return series exhibit variable order non-linear temporal dependencies, as well as dependence jumps.

Our experimental results provided strong evidence that our method is actually capable of delivering on its goals, allowing for obtaining much better performance compared to: (i) baseline (first-order) HMMs; (ii) the HMM$^{\infty}$ model, designed for capturing arbitrarily long temporal dependencies; and (iii) state-of-the-art methods in the considered application domains, e.g. methods belonging to the GARCH family. As we showed, these encouraging performance results come at a very low additional computational cost compared to existing approaches; thus, our method offers a favourable performance/complexity trade-off.

An issue we have not fully addressed in this work is how we could allow for automatic determination of the optimal model configuration, without the need of resorting to cross-validation (as we did in our experimental evaluations). For this purpose, one could resort to devising a nonparametric Bayesian construction for the VDJ-HMM model, by imposing appropriate priors over the model parameters (e.g., Dirichlet process priors [175] over the transition probability matrices of our model), and performing Bayesian inference instead of maximum-likelihood training. This issue remains to be addressed in our future work.

6.1.2 Student-t HMM credit rating system

In chapter 5, we proposed a novel credit rating system, leveraging the attractive properties of SHMMs. Our proposed approach is a holistic corporate credit scoring system, that addresses all the parts of the modeling pipeline, from financial ratio time-series selection and pre-processing, to selection of appropriate time-series modeling techniques, and information fusion strategies used to obtain the final credit scores. The core modeling stage of our system constitutes a novel financial time-series modeling scheme based on SHMMs. The utilization of SHMMs allows for capturing intricate temporal dynamics in the modeled data, reflecting the evolution of corporate behaviour and risk depending on the latent state of the economy. Furthermore, for the first time in the related literature, we employ an HMM using multivariate Student's-t mixture models as its state emission distributions. This selection allows for us to obtain a model training algorithm with high robustness to outliers in the observed datasets, which constitute a common problem in financial time-series data; in addition, it also allows for better capturing correlations between the modeled financial ratios. Finally, our model obviates the need of resorting to the information-wasting observed variable quantization procedures related approaches require, which in turn offers increased robustness to overfitting.

We performed extensive experimental evaluations of our approach using data from the Central Bank of Greece that pertain to both SMEs and large corporations, recorded over the period 2006-2012. As we showed, our approach consistently outperforms a series of benchmark approaches, both in terms of the obtained GINI coefficients and K-S statistics, and in terms of the obtained predictive variance, which quantifies the model's capacity to retain the high performance levels observed in the in-sample dataset when evaluation is performed using out-of-sample and out-of-time datasets. This performance consistency implies a much stronger generalization capacity compared to the state-of-the-art, which renders our approach much more attractive to researchers and practitioners working in real-world financial institutions, who are mainly interested in the generalization capacity of their systems, rather than in their in-sample performance. Finally, as a concluding note, we underline that both training and prediction generation using our system are extremely efficient and scalable to large datasets, without noteworthy overheads compared to existing benchmark systems. Thus, our system does not bring to the fore any kind of trade-off between computational complexity and predictive performance.

One aspect that this work did not consider is whether allowing for our model to account for skewness in the observed data could result in yielding even better predictive performances. For example, for this purpose we could consider using mixtures of multivariate skewed-t distributions as the postulated emission distributions [176; 177] instead of simple multivariate Student's-t distributions. However, the trade-offs between the obtained predictive performance increase and the increased computational costs resulting from such a modeling selection must be thoroughly examined. Further, exploration of Bayesian inference techniques for our model, which could allow for better accounting for uncertainty in the modeled data, is also worth of investigation. Finally, we also consider possible extensions of our model to allow for embedding expert judgments as an overlay, macroeconomic covariates, or covariates pertaining to qualitative information that credit officers may get aware of before it becomes depicted in the balance sheets of the examined companies. Such modifications could allow for addressing one of the main criticisms against scoring models, regarding their inability to capture rapid changes in corporation state that cannot be immediately reflected in their balance sheets.

Finally, as an aside we note that in our approach we have postulated univariate component SHMMs, modeling each financial ratio independently of all the others. Even though this modeling selection may result in not allowing for the employed dynamic models (SHHMs) to extract salient covariances information, it also true that it offers a set of significant merits in the context of our approach: Specifically, it protects our system from problems arising from multi-colinearities in the modeled data, and it prevents overfitting, which is likely to occur when modeling high-dimensional data. Nevertheless, exploring the utility of equivalent multivariate SHMM formulations in the context of our system remains an interesting question we intent to explore in our future work.

6.1.3 Other topics worth exploring regarding HMM in finance

Other interesting issues around the application of HMM in finance not addressed in the current literature and not exploited in this thesis but worth investigating in the future are listed below

6.1.3.1 Maximum weighted likelihood Estimation

The maximum likelihood estimation approach allocates equal weight in each observation of a sequence. There is often the case though that analyst needs to allocate different weights in each point in time due to different level of confidence for the information included in the respective value. For example a variable is measured with a different level of noise/error which is known. Then it is possible to adjust the estimation of the parameters of HMMs using different weights for each time stamp to account for this. Another use is the case were recent observations are considered more important and we want to allocate more weight to them in the estimation process. The Maximum weighted likelihood function is given by

$$\mathbf{L} = \prod_{i=1}^{n} \mathbf{p}(\mathbf{x}_\mathrm{i};\boldsymbol{\theta})^{\mathrm{w_i}}$$

Hu and Zidek analyze the MWL estimation in [178]. In appendix 3 an extension of the EM for the GHMM is outlined producing MWL estimation for the parameters.

6.1.3.2 Mixed Mixture Components from different distribution families

In the current thesis and in most studies using HMM the conditional distribution family is symmetric for each hidden state and its component belong to the same family of distributions. Although Gaussian and Student-t mixture enhance distribution fitting efficiency many real life dataset especially in finance exhibit distributional properties which may be better capture using an non homogeneous mixture components for example Gaussian and exponential or student-t and Gaussian or inverse Gaussian, Skewed Student-t and Gaussian components. This variation is probable to offer an increase in the ability of HMM to fit financial time series. Their estimation is relevant easy since the EM algorithm can be extended to calculate the components weights for each mixture taking into account the different distribution family. For example

$$\gamma_t(j,m) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{n}\alpha_t(j)\beta_t(j)}\right]\left[\frac{w_{jm}t\left(v_{jm},o_t,\mu_{jm},\Sigma_{jm}\right)}{\sum_{k=1}^{M}w_{jk}t\left(v_{jk},o_t,\mu_{jk},\Sigma_{jk}\right)+\sum_{\boldsymbol{k=1}}^{\boldsymbol{M}}\boldsymbol{w_{jk}}\mathbf{N}\left(\boldsymbol{o_t},\boldsymbol{\mu_{jk}},\boldsymbol{\Sigma_{jk}}\right)}\right] \quad (3)$$

extends the probability estimation that an observation o_t was produced by Student-t m component at the hidden state j in a HMM set up where each conditional mixture includes both student-t and Gaussian components. Another example of a mixture with components from different families can be found in [179].

6.1.3.3 Multivariate VaR under an HMM framework

HMM have been applied for estimating univariate VaR in [180] and produced satisfactory results. The multivariate case though to our knowledge has not been investigated in the literature. Based on theory, HMM with mixture distributions are efficiently extended to analyzing multivariate time series. Under this approach asset correlation can be directly modeled and Value at risk estimation can be robustly enhanced.

As an overall conclusion for our research study if that HMM offer a powerful framework for financial time series modeling, with competitive performance in many real life problems against well established statistical techniques. Moreover based on the current application and studies around Hidden Markov Models it is evident that HMM capabilities have not been fully explored in the current academic literature for addressing problems in the field of computational finance.

Appendix 1: Estimation Algorithms for VDJ HMM

Definition of VDJ-HMM

Let's denote by o_t the observation sequence, q_t the hidden states affecting the observation and z_t a hidden Markov process which sets the order on q_t process. In our setup q_t is not a Markov process since the Markov property is relaxed and the process exhibits jumps in the order affecting its evolution.

Based on the graph we have the following setup:

Let λbe the maximum order of the structure.

$\boldsymbol{\pi_\lambda}$. The initial distribution of the Markov process that defines the evolution of the order of the hidden process

$\boldsymbol{\pi_q}$.The initial distribution of the hidden process affecting the observation process. It is the distribution of the first λ-ple of the $q_i, i = 1, \dots, \lambda$ and we can introduce parameters for this but the number needed will be large. Instead we can distribute the same probability in occurring of each λ-ple so we set $\pi_q(q_1 = i_1, \dots q_\lambda = i_\lambda) = \frac{1}{N^\lambda}$

Let's denote by Aa matrix describing the transition probabilities of process q_t. And

$$\mathrm{A}_{ij}^{k} = P[q_t = j \mid q_{t-1} = i_1, \dots, q_{t-\lambda} = i_\lambda \;, z_t = k] = P[q_t = j \mid q_{t-k} = i_k]$$

k= 1,…,λ

The A matrix can either not depend on k so $\mathrm{A}_{ji} = P[q_t = j \mid q_{t-1} = i_1, \dots, q_{t-\lambda} = i_\lambda \;, z_t = k] = P[q_t = j \mid q_{t-k} = i]$ For k =1 , …,λ A_{ji}

or can depend on k where in this case we will have k different A^k matrices.

The conditional distribution corresponding to each hidden state can be either continuous (Gaussian, Gaussian Mixture, Student-t) or discrete

Where j have discrete values from 1… N

$$b_j(h) = P[o_t = h \mid q_t = j]$$

As mentioned above process z_t is a Markov process not observed with discrete values 1…λ. This means that the maximum order modeled through this novel approach is λ.

$$\gamma_{ij} = P[z_{t+1} = j \mid z_t = i]$$

In the γ_{ij} transition matrix if the probability γ_{11} is close to 1 then the process q_t is almost a first order Markov process. In addition for λ=1 we get the first order hidden Markov model. At the same time with λ>1 we can model higher order time series.

Conditioned on z_{t-1}, q_t the first layer process becomes a Markov process because

$P[q_t = j \mid q_{t-1} = i_1, \ldots, q_{t-\lambda} = i_\lambda \ , z_{t-1} = k] = \sum_{h=1}^{\lambda} P[q_t = j \mid q_{t-h} = i_h] * P[z_t = h \mid z_{t-1} = k]$.

(Raftery 1985, A model for high order Markov chains)

The estimation for this novel structure is accomplished using the EM algorithm.

Expectation – Maximization (Baum –Welch)

The Q-function of this structure

$$Q(\Lambda, \Lambda') = \sum_{z,q} logP(O_1^T, q, z \mid \Lambda) * P(O_1^T, q, z \mid \Lambda')$$

Let the observation sequence O_1^T and initialization of the parameters Λ (A, B, Γ, π_z,π_q)

Then based on the structure and the conditional independence stemming from the graph we get

$$P(O_1^T, z_\lambda^T, q_1^T|\Lambda) = \underbrace{P(z_\lambda|\Lambda) * P(q_1^\lambda|\Lambda)}_{} * \prod_{t=\lambda+1}^{T} \underbrace{P(z_t|z_{t-1}, \Lambda)}_{} * \prod_{t=\lambda+1}^{T} \underbrace{P(q_t|z_t, q_{t-z_t}, \Lambda)}_{} * \prod_{t=1}^{T} \underbrace{b_{q_t}(o_t|\Lambda)}_{}$$

(1)

This is the probability of the joint event of observing the sequence O_1^T and at the same time a specific realization of z_λ^T, q_1^T. if we sum all possible q, z we estimate the $P(O_1^T|\Lambda)$ under the specific group of parameters Λ. Based on our initial specification the following hold:

$$P(z_\lambda|\Lambda) = \pi_{z_\lambda}$$

$$P(z_t|z_{t-1}, \Lambda) = \gamma_{z_{t-1}z_t}$$

$$P(q_t|z_t, q_{t-z_t}, \Lambda) = a_{q_{t-z_t}q_t}$$

$$P(O_1^T|z_\lambda^T, q_1^T, \Lambda) = \prod_{t=1}^{T} b_{q_t}(o_t|\Lambda)$$

$$P(q_1^T, |z_\lambda^T, \Lambda) = \frac{1}{N^\lambda} \prod_{t=\lambda+1}^{T} a_{q_{t-z_t}q_t}$$

$$P(z_\lambda^T|\Lambda) = \pi_{z_\lambda} \prod_{t=\lambda+1}^{T} \gamma_{z_{t-1}z_t}$$

Since,

$$P(O_1^T, z_\lambda^T, q_1^T|\Lambda) = \underbrace{P(z_\lambda|\Lambda) * P(q_1^\lambda|\Lambda)}_{} * \prod_{t=\lambda+1}^{T} \underbrace{P(z_t|z_{t-1}, \Lambda)}_{} * \prod_{t=\lambda+1}^{T} \underbrace{P(q_t|z_t, q_{t-z_t}, \Lambda)}_{} * \prod_{t=1}^{T} \underbrace{b_{q_t}(o_t|\Lambda)}_{}$$

by taking the log in the above relationship and summing over all possible combinations of the hidden variables sequences the q- function becomes

$Q(\Lambda, \Lambda') = \sum_{z,q} log P(z_\lambda|\Lambda) * P(O_1^T, q, z\,|\Lambda')\ (A)+$

$\sum_{z,q} log P(q_1^\lambda\,|\Lambda) *$

$P(O_1^T, q, z \mid \Lambda')(B)\sum_{z,q} \sum_{t=\lambda+1}^{T} log P(z_t|z_{t-1}, \Lambda) * P(O_1^T, q, z \mid \Lambda')$ (C)+$\sum_{z,q} \sum_{t=\lambda+1}^{T} log P\left(q_t \middle| z_t, q_{t-z_t}, \Lambda\right) *$ $P(O_1^T, q, z \mid \Lambda')$ (D)+$\sum_{z,q} \sum_{t=1}^{T} log b_{q_t}(o_t|\Lambda) * P(O_1^T, q, z \mid \Lambda')$ (E)

For each component (A) to (E) we sum over all possible combinations of z, q hidden sequences and the variables not included in the right hand side disappear and the marginal probabilities of the remaining variables are formed:

A =>$\sum_{z,q} log P(z_\lambda|\Lambda) * P(O_1^T, q, z \mid \Lambda') = \sum_{i=1}^{\lambda} log \pi_\lambda(i) * P(O_1^T, z_\lambda = i \mid \Lambda')$

$\frac{d(\sum_{i=1}^{\lambda} log \pi_\lambda(i) * P(O_1^T, z_\lambda = i \mid \Lambda') + \gamma * (\sum_{i=1}^{\lambda} \pi_\lambda(i) - 1)}{d\pi_\lambda(i)} = 0$ =>$\frac{1}{\pi_\lambda(i)}$* $P(O_1^T, z_\lambda = i \mid \Lambda') + \gamma = 0$ =>

$\sum_{i=1}^{\lambda} P(O_1^T, z_\lambda = i \mid \Lambda') + \gamma * (\sum_{i=1}^{\lambda} \pi_\lambda(i) - 1) = 0$ =>γ=- $P(O_1^T|\Lambda')$*so*

$$\pi_\lambda(i) = \frac{P(O_1^T, z_\lambda = i \mid \Lambda')}{P(O_1^T|\Lambda')} \text{ (2)}$$

B =>$\sum_{z,q} log P(q_1^\lambda \mid \Lambda) * P(O_1^T, q, z \mid \Lambda')$ since this is the initial set to be$P\left(q_1^\lambda \middle| \Lambda\right) = \frac{1}{N^P}$ no need to maximize it

C =>$\sum_{z,q} \sum_{t=\lambda+1}^{T} log P(z_t|z_{t-1}, \Lambda) * P(O_1^T, q, z \mid \Lambda') = \sum_{j=1}^{\lambda} \sum_{i=1}^{\lambda} \sum_{t=\lambda+1}^{T} log P(z_t = i \mid z_{t-1} = j, \Lambda) *$ $P(O_1^T, q, z \mid \Lambda') = \sum_{j=1}^{\lambda} \sum_{i=1}^{\lambda} \sum_{t=\lambda+1}^{T} log \gamma_{ji} * P(O_1^T, z_t = i, z_{t-1} = j|\Lambda')$

$$P(O_1^T, z_t = i \mid \Lambda') = \sum_{j=1}^{\lambda} P(O_1^T, z_t = i, z_{t-1} = j|\Lambda')$$

$$\frac{d(\sum_{j=1}^{\lambda} \sum_{i=1}^{\lambda} \sum_{t=\lambda+1}^{T} log \gamma_{ji} * P(O_1^T, z_t = i, z_{t-1} = j|\Lambda') + \sum_{j=1}^{\lambda} s_j(\sum_{i=1}^{\lambda} \gamma_{ji} - 1))}{d\gamma_{ji}} = 0$$

For specific j we get

$$\frac{d(\sum_{i=1}^{\lambda}\sum_{t=\lambda+1}^{T} log\gamma_{ji} * P(O_1^T, z_t = i,\ z_{t-1} = j|\Lambda') +\ s_j(\sum_{i=1}^{\lambda}\gamma_{ji} - 1))}{d\gamma_{ji}} = 0$$

By differentiating for all i with constant j and summing for all i we get

$$s_j * \ \sum_{i=1}^{\lambda}\gamma_{ji} + \sum_{i=1}^{\lambda}\sum_{t=\lambda+1}^{T} P(O_1^T, z_t = i,\ z_{t-1} = j|\Lambda') = 0 \Rightarrow s_j = -\sum_{t=\lambda+1}^{T} P(O_1^T,\ z_{t-1} = j|\Lambda')$$

$$\gamma_{ji} = \frac{\sum_{t=\lambda+1}^{T} P(O_1^T, z_t = i,\ z_{t-1} = j|\Lambda')}{\sum_{t=\lambda+1}^{T} P(O_1^T,\ z_{t-1} = j|\Lambda')} \ , \forall\ i, j \ (3)$$

$$D \Rightarrow \sum_{z,q}\sum_{t=\lambda+1}^{T} log P(q_t|z_t, q_{t-z_t}, \Lambda) * P(O_1^T, q, z\ |\Lambda') = \sum_{k=1}^{\lambda}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=\lambda+1}^{T} log P(q_t = i, z_t = k, q_{t-k} = j\ |\Lambda) * P(O_1^T, q_t = i, z_t = k, q_{t-k} = j|\Lambda') = \sum_{k=1}^{\lambda}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=\lambda+1}^{T} log A_{ji}^{(\kappa)} * P(O_1^T, q_t = i, z_t = k, q_{t-k} = j|\Lambda')$$

The k in the parenthesis indicates a different transition matrix for different values of k – lag (if preferred for modeling purposes)

By differentiating we get

$$A_{ji} = \frac{\sum_{k=1}^{\lambda}\sum_{t=\lambda+1}^{T} P(O_1^T, q_t = i, z_t = k, q_{t-k} = j\ |\Lambda')}{\sum_{k=1}^{\lambda}\sum_{t=\lambda+1}^{T} P(O_1^T, z_t = k, q_{t-k} = j\ |\Lambda')} \ (4)$$

togetadifferenttransitionprobabilitybykorderwemayusethefollowingformula.

$$A_{ji}^{(k)} = \frac{\sum_{t=\lambda+1}^{T} P\left(O_1^T, q_t = i, z_t = k, q_{t-k} = j\middle|\Lambda'\right)}{\sum_{t=\lambda+1}^{T} P\left(O_1^T, z_t = k, q_{t-k} = j\middle|\Lambda'\right)}$$

The state-conditional density distribution of the observed sequence can either be continuous or discrete-valued and is estimated maximizing relationship (E). In our implementation we assume Gaussian mixtures distributions as the conditional densities distributions for each hidden state. Irrespective of the structure of the hidden stochastic process the relationship (conditional dependence) with the observe process remains the same as the usual structure. For Gaussian Mixtures, the form of the Q-function is slightly different, i.e., the hidden variables must include not only the hidden state sequence, but also a variable indicating the mixture component for each state at each time.

Lets denote by

$$b_{jm}(o_t) \sim N\left(o_t, \mu_{jm}, \Sigma_{jm}\right) = \frac{1}{|\Sigma_{jm}|^{\frac{1}{2}}(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}(o_t-\mu_{jm})^T \Sigma_{jm}^{-1}(o_t-\mu_{jm})}$$

the m-component of the Gaussian mixture given that the hidden state is j (total components in the mixture c)

and summing over m we get $b_j(o_t) = P[o_t \mid q_t = j] = \sum_{m=1}^{c} w_{jm} * b_{jm}(o_t)$

We denote$\gamma_{c_t}(j,m) = P(q_t = j, c_t = m \ |\Lambda', O_1^T)$ the probability at time t the hidden state equals j and the component that produced the observation is the m-th component of the j-th mixture, where $c_t$is the component at time t

$\gamma_t(i)= P(q_t = j|\Lambda', O_1^T)$ the probability at time t the hidden state equals j

Due to the conditional independence embedded in the structure of the hmm we get

$$P(q_t = j, c_t = m \ |O_1^T, \Lambda') = P(c_t = m \ |O_1^T, q_t = j, \Lambda') * P(q_t = j|O_1^T, \Lambda')$$

So based on the above we get

$$\gamma_{c_t}(j,m) = \gamma_t(i) * \left[\frac{w_{jm}N\left(o_t,\mu_{jm},\Sigma_{jm}\right)}{\sum_{k=1}^{M} w_{jk}N\left(o_t,\mu_{jk},\Sigma_{jk}\right)}\right](5)$$

After the estimation of $\gamma_{c_t}(j,m)$ the means, standard deviations and weights for each component are calculated using (6), (7), (8) below and the derivations of these relationships are the same with every other hmm. For more details see [181],

Weights:$w_{jm} = \frac{\sum_{t=1}^{T} \gamma_{c_t}(j,m)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_{c_t}(j,k)}$(6)

Mean: $\mu_{jm} = \frac{\sum_{t=1}^{T} \gamma_{c_t}(j,m) o_t}{\sum_{t=1}^{T} \gamma_{c_t}(j,m)}$(7)

Standard deviation:$\Sigma_{jm} = \frac{\sum_{t=1}^{T} \gamma_{c_t}(j,m)\left(o_t - \mu_{jm}^{new}\right)\left(o_t - \mu_{jm}^{new}\right)^T}{\sum_{t=1}^{T} \gamma_{c_t}(j,m)}$(8)

To sum up in order to recalculate the variables in each iteration step of the Expectation Maximization Algorithm we have to estimate the following probabilities:

$\gamma_t(i) = P\ (q_t = j | O_1^T, \Lambda'),$ for t=1,...,T

$\gamma_{c_t}(j,m) = P(O_1^T, q_t = j\ , c_t = m\ \ |\Lambda'),$ for t=1,...,T

$\gamma_{zqq_t}(j,i,k) = P(O_1^T, q_t = i, z_t = k, q_{t-k} = j\ |\Lambda'),$ for t=λ+1,...,T

$\gamma_{zq_t}(j,k) = P(O_1^T, z_t = k, q_{t-k} = j\ |\Lambda')$,for t=λ+1,...,T

$\gamma_{zz_t}(j,i) =\ P(O_1^T, z_t = i,\ z_{t-1} = j|\Lambda')$,for t=λ+1,...,T

$\gamma_{z_t}(j) = P(O_1^T,\ z_t = j|\Lambda')$,for t=λ+1,...,T

In order to estimate the parameters of the model we need to define the respective a-pass and b-pass algorithms for the new structure to perform our calculations. In order to proceed with the estimation methodology we have to deal with the starting parameters, and to define the necessary forward a-pass and backward b-pass probabilities. Since the order of q_t is λwe denote the initial state is Π(q$_1$, …,q$_λ$).

This is the distribution of the first λ hidden states. In order to avoid introducing more parameters into the system we assume that Π(q1, …,qλ) is uniformly distributed meaning that

$P(\, q_1, \dots, q_\lambda | \Lambda') = \frac{1}{N^\lambda}$, and independent of z_λ.

Forward Algorithm

We define the a pass by

$\boldsymbol{a_t(i_1, \dots, i_\lambda, k)} = P(q_t \ = \ i_1, \dots, q_{t-\lambda+1} = i_\lambda, z_t \ = \ k, O_1^T | \Lambda')$

Initialize

$\boldsymbol{a_\lambda(i_1, \dots, i_\lambda, k)} = P(q_1 \ = \ i_1, \dots, q_\lambda = i_\lambda, z_\lambda \ = \ k, O_1^\lambda | \Lambda') =$

$P(z_\lambda \ = \ k|\Lambda') * P(q_1 \ = \ i_1, \dots, q_\lambda = i_\lambda | \Lambda', z_\lambda \ = \ k) \ *P\left(O_1^\lambda \middle| \Lambda', q_1 \ = \ i_1, \dots, q_\lambda = i_\lambda \,\right)$

$= \pi_\lambda(k) * \frac{1}{N^\lambda} * \prod_{t=1}^{\lambda} b_{q_t}\,(o_t).$

Induction Step

$\boldsymbol{a_t\ (q_{t-\lambda+1}^t, z_t)} = \boldsymbol{a_t\ (i_1, \dots, i_\lambda, k)} = P(q_t \ = \ i_1, \dots, q_{t-\lambda+1} = i_\lambda, z_t \ = \ k, O_1^t | \Lambda') =$

$P(O_t | O_1^{t-1}, q_{t-\lambda+1}^t, z_t, \Lambda') * P(O_1^{t-1}, q_{t-\lambda+1}^t, z_t = k | \Lambda') = P(O_t | q_t, \Lambda') * P(O_1^{t-1}, q_{t-\lambda+1}^t, z_t = k | \Lambda') =$

$P(O_t | q_t, \Lambda') * \sum_{j=1}^N \sum_{i=1}^\lambda P(O_1^{t-1}, q_{t-\lambda+1}^t, z_t = k, z_{t-1} = i, q_{t-\lambda} = j \ | \Lambda') =$

$P(O_t | q_t, \Lambda') * \sum_{j=1}^N \sum_{i=1}^\lambda P(z_t = k \middle| O_1^{t-1}, q_{t-\lambda+1}^t \,, z_{t-1} = i, q_{t-\lambda} = j \,, \Lambda') \ * P\big(q_t | O_1^{t-1}, q_{t-\lambda+1}^{t-1}, z_{t-1,} =$

$i, z_t = k, q_{t-\lambda} = j \,, \Lambda'\big) \ * \ P(O_1^{t-1}, q_{t-\lambda+1}^{t-1}, z_{t-1} = i, q_{t-\lambda} = j \,, \Lambda') =$

$P(O_t | q_t, \Lambda') \ * \sum_{j=1}^N \sum_{i=1}^\lambda P(z_t = k | z_{t-1} = i, \Lambda') * P\big(q_t | q_{t-\lambda+1}^{t-1}, z_t = k, q_{t-\lambda} = j \,, \Lambda'\big) * a_{t-1}$

$(q_{t-\lambda}^{t-1}, z_{t-1}) =$

$$b_{q_t}(o_t|\Lambda) *\sum_{j=1}^{N}\sum_{i=1}^{\lambda}\gamma_{z_t z_{t-1}\underbrace{=i * P\big(q_t|q_{t}}_{}-z_t},\Lambda'\big) * a_{t-1}\ (q_{t-\lambda+1}^{t-1}, q_{t-\lambda}=j, z_{t-1}=i)$$

$$if\ z_t=\lambda\ then\ P(q_t|q_{t-\lambda}=j\ ,\Lambda')$$

$$b_{q_t}(o_t|\Lambda) *\sum_{j=1}^{N}\sum_{i=1}^{\lambda}\gamma_{ik} * P(q_t|q_{t-k},\Lambda') * a_{t-1}\ (q_{t-\lambda+1}^{t-1}, q_{t-\lambda}=j, z_{t-1}=i)$$

Where$\boldsymbol{P(q_t|q_{t-k},\Lambda') = A_{i_1 i_{k+1}}^{(k+1)}}$

Backward Algorithm

We define the β pass by

$\boldsymbol{\beta_t(i_1,\dots,i_\lambda,k)} = P(O_{t+1}^T \mid q_t = i_1,\dots,q_{t-\lambda+1}=i_\lambda, z_t = k\ |\Lambda')$

Initialize

$\boldsymbol{\beta_1(i_1,\dots,i_\lambda,k)} = 1\ \forall\ 1 \le i_1,\dots,i_\lambda \le N\ ,\ 1 \le k \le \lambda$

Induction Step

Let$\lambda \le t \le T$

Then, $\boldsymbol{\beta_t(i_1,\dots,i_\lambda,k)} = P(O_{t+1}^T \mid q_t = i_1,\dots,q_{t-\lambda+1}=i_\lambda, z_t = k\ |\Lambda') = \sum_{j=1}^{N}\sum_{s=1}^{\lambda}P(O_{t+1}^T\ ,\ q_{t+1} = j,\ z_{t+1} = s \mid q_t = i_1,\dots,q_{t-\lambda+1}=i_\lambda, z_t = k\ |\Lambda') =$

$\sum_{j=1}^{N}\sum_{s=1}^{\lambda}P(O_{t+1} \mid O_{t+2}^T,\ q_{t+1}=j,\ z_{t+1} = s,\ q_t = i_1,\dots,q_{t-\lambda+1}=i_\lambda, z_t = k\ ,\Lambda')\ *$
$P(O_{t+2}^T, q_{t+1}=j,\ z_{t+1} = s|\ q_t = i_1,\dots,q_{t-\lambda+1}=i_\lambda, z_t = k,\Lambda') =$

$\sum_{j=1}^{N}\sum_{s=1}^{\lambda}P(\ O_{t+1} \mid\ q_{t+1}=j,\Lambda') * P(O_{t+2}^T|\ q_{t+1}=j,\ z_{t+1} = s,\ q_t = i_1,\dots,q_{t-\lambda+1}=i_\lambda, z_t = k,\Lambda')\ *\ P(q_{t+1}=j|\ z_{t+1} = s, q_t = i_1,\dots,q_{t-\lambda+1}=i_\lambda, z_t = k,\Lambda') *\ P(\ z_{t+1} = s\ |q_t = i_1,\dots,q_{t-\lambda+1}=i_\lambda, z_t = k,\Lambda')=$

$\sum_{j=1}^{N}\sum_{s=1}^{\lambda} b_{q_{t+1}=j}(o_{t+1}) * P(O_{t+2}^{T}|\ q_{t+1} = j,\ z_{t+1}\ =\ s,\ q_t\ =\ i_1,\dots,q_{t-\lambda+2} = i_{\lambda-1},\Lambda')\ *$
$P(q_{t+1} = j|\ z_{t+1}\ =\ s, q_t\ =\ i_1,\dots,q_{t-\lambda+1} = i_\lambda,\Lambda') *\ P(\ z_{t+1}\ =\ s\ |\ z_t\ =\ k,\Lambda')=$

$\sum_{j=1}^{N}\sum_{s=1}^{\lambda} b_{q_{t+1}=j}(o_{t+1}) * \beta_{t+1}(j, i_1,\dots,i_{\lambda-1},s) *\ P(q_{t+1} = j|\ q_{t+1-s} = i_s,\Lambda') *\ \gamma_{ks}$

Where$\boldsymbol{P(q_{t+1} = j|\ q_{t+1-s} = i_s,\Lambda')} * = \ \boldsymbol{A_{i_s i_1}^{(s)}}$

The upper script s refers to different by order transition matrix for the states

Migration Probability

We define the following probability (transition probability)

$\boldsymbol{\delta_t(i_1,\dots,i_{\lambda+1},k) = P(O_1^T, q_t\ =\ i_1,\dots,q_{t-\lambda} = i_{\lambda+1}, z_t\ =\ k\ |\Lambda')}\ =$
$P(\mathrm{O}_{\mathrm{t+1}}^{\mathrm{T}}|\ \mathrm{O}_1^{\mathrm{t}},\ \mathrm{q_t}\ =\ \mathrm{i}_1,\dots,\mathrm{q}_{\mathrm{t}-\lambda} = \mathrm{i}_{\lambda+1},\mathrm{z_t}\ =\ \mathrm{k}\ |\Lambda')\ \ * P(O_1^t, q_t\ =\ i_1,\dots,q_{t-\lambda} = i_{\lambda+1}, z_t\ =\ k\ |\Lambda') =$

$P(O_{t+1}^T|\ q_t\ =\ i_1,\dots,q_{t-\lambda+1} = i_\lambda, z_t\ =\ k,\Lambda')\ *\ \ P(O_1^t, q_t\ =\ i_1,\dots,q_{t-\lambda} = i_{\lambda+1}, z_t\ =\ k\ |\Lambda')\ =$
$\beta_t(i_1,\dots,i_\lambda,k) * P(O_t\ , O_1^{t-1}, q_t\ =\ i_1,\dots,q_{t-\lambda} = i_{\lambda+1}, z_t\ =\ k\ |\Lambda') =$

$\beta_t(i_1,\dots,i_\lambda,k) * P(O_t\ |\ \ O_1^{t-1}, q_t\ =\ i_1,\dots,q_{t-\lambda} = i_{\lambda+1}, z_t\ =\ k\ ,\Lambda') * P(\ q_t\ =\ i_1|\ O_1^{t-1}, q_{t-1}\ =$
$i_2,\dots,q_{t-\lambda} = i_{\lambda+1}, z_t\ =\ k\ ,\Lambda') * \sum_{j=1}^{\lambda} P(O_1^{t-1}, q_{t-1}\ =\ i_2,\dots,q_{t-\lambda} = i_{\lambda+1}, z_t\ =\ k,\ z_{t-1}\ =\ j|\Lambda')$

$=\beta_t(i_1,\dots,i_\lambda,k) * b_{q_t=i_1}(O_t\ ) *\ \mathrm{A}_{\mathrm{i}_{\mathrm{k+1}}\mathrm{i}_1}\ *$
$\sum_{j=1}^{\lambda} P\ (z_t\ =\ k|z_{t-1}\ =\ j,\Lambda')\ *\ \ P(O_1^{t-1}, q_{t-1}\ =\ i_2,\dots,q_{t-\lambda} = i_{\lambda+1}, z_{t-1}\ =\ j|\Lambda') =$

$$\beta_t(i_1,\dots,i_\lambda,k) * b_{q_t=i_1}(O_t\ ) *\ \mathrm{A}_{\mathrm{i}_{\mathrm{k+1}}\mathrm{i}_1}\ *\ \sum_{j=1}^{\lambda}\gamma_{jk} * a_{t-1}(i_2,\dots,i_{\lambda+1},j)$$

Note that if we assume different transition matrix by order k, $A^{(k+1)}$ then in the last equation $A^{(k+1)}_{\mathrm{i}_{k+1}\mathrm{i}_1}$

Using $\boldsymbol{\delta_t(i_1, \dots, i_{\lambda+1}, k)} = P(O_1^T,\ q_t = i_1, \dots, q_{t-\lambda} = i_{\lambda+1}, z_t = k\ |\Lambda')$we get the following relationship in a similar way

$\boldsymbol{\delta z_t(i_1, \dots, i_{\lambda+1}, k, k_1)} = P(O_1^T, q_t = i_1, \dots, q_{t-\lambda} = i_{\lambda+1}, z_t = k,\ z_{t-1} = k_1|\Lambda') = \beta_t(i_1, \dots, i_\lambda, k) * b_{q_t=i_1}(O_t) * \mathrm{A}_{\mathrm{i}_{k+1}\mathrm{i}_1} * \gamma_{k_1 k} * a_{t-1}(i_2, \dots, i_{\lambda+1}, k_1)$

Using the following probabilities $\boldsymbol{\delta_t(i_1, \dots, i_{\lambda+1}, k)}$, $\boldsymbol{\delta z_t(i_1, \dots, i_{\lambda+1}, k, k_1)}$, $\boldsymbol{a_t(i_1, \dots, i_\lambda, k)}$ and$\boldsymbol{\beta_t(i_1, \dots, i_\lambda, k)}$we can estimate the probabilities (2) – (5) in the following way

$$\gamma_t(i) = P\ (q_t = \boldsymbol{i_1}|O_1^T, \Lambda') = \frac{\boldsymbol{P\ (q_t=i_1, O_1^T|\Lambda')}}{\boldsymbol{P\ (O_1^T|\Lambda')}} = \frac{\sum_{\boldsymbol{i_2, \dots, i_\lambda}=1}^{N} \sum_{k=1}^{\lambda} \boldsymbol{a_t(i_1, \dots, i_\lambda, k) * \beta_t(i_1, \dots, i_\lambda, k)}}{\sum_{\boldsymbol{i_1, \dots, i_\lambda}=1}^{N} \sum_{k=1}^{\lambda} \boldsymbol{a_t(i_1, \dots, i_\lambda, k) * \beta_t(i_1, \dots, i_\lambda, k)}} \quad (9)$$

In addition from (5) we get $\gamma_{c_t}(j, m) = \gamma_t(i) * \left[\frac{w_{jm} N\ (o_t, \mu_{jm}, \Sigma_{jm})}{\sum_{k=1}^{M} w_{jk} N\ (o_t, \mu_{jk}, \Sigma_{jk})}\right]$

$$\boldsymbol{\gamma_{zqq_t}(j, i, k) = P\left(O_1^T, q_t = i, z_t = k, q_{t-k} = j\middle|\Lambda'\right)} = \sum_{i_2, \dots, i_k, i_{k+2}, \dots, i_{\lambda+1}=1}^{N} P\left(O_1^T, q_t = i, \dots, q_{t-k+1} = i_k, q_{t-k} = j, q_{t-k-1} = i_{k+2} \dots, q_{t-\lambda} = i_{\lambda+1}, z_t = k\ \middle|\ \Lambda'\right) = \sum_{i_2, \dots, i_k, i_{k+2}, \dots, i_{\lambda+1}=1}^{N} \delta_t\left(i, i_2, \dots, i_k, j, i_{k+2}, \dots, i_{\lambda+1}, k\right) (13)$$

$$\boldsymbol{\gamma_{zq_t}(j, k) = P\left(O_1^T, z_t = k, q_{t-k} = j\middle|\Lambda'\right)} = \sum_{i_1, \dots, i_k, i_{k+2}, \dots, i_{\lambda+1}=1}^{N} P\left(O_1^T, q_t = i_1, \dots, q_{t-k+1} = i_k, q_{t-k} = j, q_{t-k-1} = i_{k+2} \dots, q_{t-\lambda} = i_{\lambda+1}, z_t = k\ \middle|\ \Lambda'\right) = \sum_{i_1, \dots, i_k, i_{k+2}, \dots, i_{\lambda+1}=1}^{N} \delta_t\left(i_1, i_2, \dots, i_k, j, i_{k+2}, \dots, i_{\lambda+1}, k\right) (14)$$

$$\boldsymbol{\gamma_{zz_t}(j, i) = P\left(O_1^T, z_t = k,\ z_{t-1} = k1\middle|\Lambda'\right)} = \sum_{i_1, \dots, i_{\lambda+1}=1}^{N} P\left(O_1^T, q_t = i_1, \dots, q_{t-\lambda} = i_{\lambda+1}, z_t = k,\ z_{t-1} = k1\middle|\Lambda'\right) = \sum_{i_1, \dots, i_{\lambda+1}=1}^{N} \delta z_t(i_1, \dots, i_{\lambda+1}, k, k_1)\ (15)$$

$\boldsymbol{\gamma}_{\boldsymbol{z_t}}(\boldsymbol{j}) = \mathbf{P}\left(\boldsymbol{z_t} = \boldsymbol{k}, \boldsymbol{O_1^T}\middle|\boldsymbol{\Lambda}^{'}\right) =$

$$\sum_{i_1,\ldots,i_{\lambda+1}=1}^{N} P\left(O_1^T, q_t = i, \ldots, q_{t-\lambda} = i_{\lambda+1}, z_t = k \middle| \Lambda'\right) = \sum_{i_1,\ldots,i_{\lambda+1}=1}^{N} \delta_t(i_1, \ldots, i_{\lambda+1}, k,) \quad (16)$$

Based on (2) we get

$$\pi_\lambda(i) = \frac{P(O_1^T, z_\lambda = i \mid \Lambda\prime)}{P(O_1^T \mid \Lambda\prime)} = \frac{\sum_{i_1,\ldots,i_{\lambda+1}=1}^{N} \delta_\lambda(i_1,\ldots,i_{\lambda+1},k,)}{\sum_{i_1,\ldots,i_{\lambda+1}=1,k=1}^{N,\lambda} \delta_\lambda(i_1,\ldots,i_{\lambda+1},k,)} \quad (10)$$

Based on (3) we get

$$\gamma_{ji} = \frac{\sum_{t=\lambda+1}^{T} P\left(O_1^T, z_t = i, z_{t-1} = j \middle| \Lambda\prime\right)}{\sum_{t=\lambda+1}^{T} P\left(O_1^T, z_{t-1} = j \middle| \Lambda\prime\right)} = \frac{\sum_{t=\lambda+1}^{T} \boldsymbol{\gamma}_{\boldsymbol{zz_t}}(\boldsymbol{j},\boldsymbol{i})}{\sum_{t=\lambda}^{T-1} \boldsymbol{\gamma}_{\boldsymbol{z_t}}(\boldsymbol{j})} \quad (11)$$

Based on (4) we get

$$A_{ji} = \frac{\sum_{k=1}^{\lambda}\sum_{t=\lambda+1}^{T} P\left(O_1^T, q_t = i, z_t = k, q_{t-k} = j \mid \Lambda\prime\right)}{\sum_{k=1}^{\lambda}\sum_{t=\lambda+1}^{T} P\left(O_1^T, z_t = k, q_{t-k} = j \mid \Lambda\prime\right)} = \frac{\sum_{k=1}^{\lambda}\sum_{t=\lambda}^{T} \gamma_{zqq_t}(j,i,k)}{\sum_{k=1}^{\lambda}\sum_{t=\lambda+1}^{T} \gamma_{zq_t}(j,k)} \quad (12)$$

EM Algorithm for Solving VDJ-HMM

The following equations summarize the estimation procedure of the parameters of VDJ-HMM.

Iterations Counter: k: = 0

1. Set the initial values of the parameters of an VDJ-HMM.
2. Calculation using the forward - backward algorithm of $\boldsymbol{\alpha_t(i_1, \ldots, i_\lambda, k)}$, $\boldsymbol{\beta_t(i_1, \ldots, i_\lambda, k)}$, $\boldsymbol{\delta_t(i_1, \ldots, i_{\lambda+1}, k)}$, $\boldsymbol{\delta z_t(i_1, \ldots, i_{\lambda+1}, k, k_1)}$variables
3. E-step: calculation of the variables $\gamma_t(i), \gamma_{c_t}(j,m), \gamma_{zqq_t}(j,i,k), \gamma_{zq_t}(j,k), \gamma_{zz_t}(j,i), \gamma_{z_t}(j)$using the equations(5), (9), (13),(14),(15),(16) respectively
4. M-step: Calculation of variables $\boldsymbol{\Sigma_{jm}}$, $\boldsymbol{\mu_{jm}}$, $\boldsymbol{w_{jm}}$, $\boldsymbol{\pi_\lambda}$, $A_{ji}^{(k)}$, $\boldsymbol{\gamma_{ji}}$usingtheequations(6), (7), (8), (10), (12) ,(11) respectively
5. If EM algorithm converges ie**l − l'**< threshold then finish else increase by 1 the iteration counter k: = k + 1 and repeat steps 2-5.

Log-likelihood Estimation

Equation (6) describes the joint probability of$O_1^T$with a specific realization of z_λ^T, q_1^Thidden variables. If we sum over all z_λ^T, q_1^Thidden sequences we get the $P(O_1^T|\Lambda)$ under the specific parameterization. The likelihood of the observed sequence of length T ($P(O_1^T|\Lambda)$) is, as usual, estimated using the forward algorithm described previously by summing over all possible values of the λ+1 ple, q_T , ... $q_{T-\lambda+1}, z_T$

$$P(O_1^T|\Lambda') = \sum_{i_1,\ldots,i_\lambda=1,k=1}^{N,\lambda} P(q_T = i_1, \ldots, q_{T-\lambda+1} = i_\lambda, z_T = k, O_1^T |\Lambda') = \sum_{i_1,\ldots,i_\lambda=1,k=1}^{N,\lambda} a_T (i_1, \ldots, i_\lambda, k)$$

Appendix 2: Expectation Maximization Algorithm

A popular method for estimating the parameters of an HMM is Baum - Welch Algorithm [182], a technique where in the case of hidden Markov models coincides with the solution of the system using the well - known Expectation Maximization algorithm [6]. It should be noted that the solving method developed by Baum and his colleagues preceded the theoretical occurrence of EM by 10 years. Initially, the EM algorithm has been developed for estimating the maximum likelihood in the case of systems estimation with incomplete data. However, its use was expanded in the case of latent variables like the theory of HMM. For example in the case of GHMM for one length T sequence the hidden variables are the system states and the sequence that determines the component distribution of the mixture from which visible variable was produced. Today the Expectation Maximization algorithm is one of the most widespread methods for estimating maximum likelihood. This appendix summarizes the basic principles of the general form of the algorithm applied to a series of problems nowadays that contain hidden / incomplete data.

The EM algorithm is a general method to estimate maximum likelihood for problems with incomplete data. From conception onwards has been applied to a variety of cases from the estimation of mixed probability distributions to the voice recognition problems and tomography. There are two main applications of the algorithm EM. The first is the data actually are missing values, due to problems and limitations in the observation process, while the second in optimizing the likelihood function is difficult to solve analytically, while assuming the existence of additional variables which are either missing (or hidden) can simplify the estimation process of the MLE. The second application is more common in computational pattern recognition community.

Suppose that dataset X is observed and follow a specific distribution. Let us denote by Y the set of data that is not observed. We assume that the full data set Z = (X, Y) have joint density function given by the following equation

$$p(z|\Theta) = p(x,y|\Theta) = p(y|\ x,\Theta)p(x|\Theta)\ (1)$$

.

Because the dataset Y traditional methods to estimate the parameters maximizing the likelihood function is not possible. The algorithm EM, makes use of an auxiliary function, the so-called "Q-function", to deal with missing data. Essentially the EM initially estimates the expected value of the logarithm of the likelihood function for the data set based on all possible values of Y given the price of the set X and some initial estimates of the parameters in the function f(y| X, $\Theta^{(i-1)}$). Therefore the Q-function is defined as follows:

$$Q(\Theta, \Theta^{(i-1)}) = \int_{y \in Y} log p(X, y|\Theta) f(\ y|\ X, \Theta^{(i-1)}) dy$$

Where $\Theta^{(i-1)}$are current estimates of the system variables used to calculate the expected value andΘ are the new values of the system parameters calculated by the function Q through maximization. Specifically the function (2) X, $\Theta^{(i-1)}$are considered known, Θ is the set of values of the parameters to be estimated and Y is a set of random variables defined by the function f(y| X, $\Theta^{(i-1)}$). Therefore equation 2 can be rewritten as follows:

$$Q(\Theta, \Theta^{(i-1)}) = \int_{y \in Y} log p(X, y|\Theta) f(\ y|\ X, \Theta^{(i-1)}) dy$$

The function f(y| X, $\Theta^{(i-1)}$)is the marginal distribution of the latent data and depend only on the values of the data set X and the current values of the parameters. The estimation of the expected value of the function is the so-called E-step of EM. From the definition of Q function Θ refers to the set of parameters to be estimated in order to maximize the likelihood function while $\Theta^{(i-1)}$correspond to the parameters we use to estimate the expected value of the function in order to substitute the values of hidden variables with their respective expected values. The second step of the EM algorithm, the M-Step involves the maximizing of Q function with respect to Θ. That is

$$\Theta^{(i)} = \underset{\Theta}{argmax}\, Q(\Theta, \Theta^{(i-1)})$$

These two steps are repeated in the same order. Each iteration has been proven to lead to greater or equal price of the log-likelihood function and the algorithm by default converges at least to a local maximum of the likelihood function. For a detailed presentation of properties and theoretical evidence of the algorithm please refer to [6].

Appendix 3: Maximum weighted likelihood for GHMM

EM for Maximum Weighted Likelihood

In the general case the log weighted likelihood can be written as follows

$$\mathrm{l}=\sum_{i} \mathrm{w_i}\log\left(\mathrm{p}(\mathrm{x_i};\theta)\right)$$

όπου$\sum_{\mathrm{i}} \mathrm{w_i}= 1$

Thus the following hold for the log function

$$\mathrm{l} =\sum_{\mathrm{i}} \mathrm{w_i}\log\left(\sum_{\mathrm{z_i}} \mathrm{p}(\mathrm{x_i},\mathrm{z_i};\theta)\frac{\mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)}{\mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)}\right)$$

$$\mathrm{l} =\sum_{\mathrm{i}} \mathrm{w_i}\log\left(\sum_{\mathrm{z_i}} \mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)\frac{\mathrm{p}(\mathrm{x_i},\mathrm{z_i};\theta)}{\mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)}\right)>=\sum_{\mathrm{i}} \mathrm{w_i}\sum_{\mathrm{z_i}} \mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)\log\frac{\mathrm{p}(\mathrm{x_i},\mathrm{z_i};\theta)}{\mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)}$$

$$=\sum_{\mathrm{i}} \mathrm{w_i}\sum_{\mathrm{z_i}} \mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)\log\frac{\mathrm{p}(\mathrm{x_i},\mathrm{z_i};\theta)}{\mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)}$$

$$\Theta = \mathrm{argmax}\left(\sum_{\mathrm{i}} \mathrm{w_i}\sum_{\mathrm{z_i}} \mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)\log \mathrm{p}(\mathrm{x_i},\mathrm{z_i};\theta)\right) = \mathrm{argmax}\left(\sum_{\mathrm{i}}\sum_{\mathrm{z_i}} \mathrm{p}(\mathrm{z_i}|\mathrm{x_i};\theta_t)\mathrm{w_i}\log \mathrm{p}(\mathrm{x_i},\mathrm{z_i};\theta)\right)$$

Thus the weights appear in as a product for each observation in the Q function of the log weighted likelihood.

The EM for the MWL estimation for the GHMM

Based on the above result in the case of weighted likelihood with weight scheme

δ_t , t=1,…, T Q function becomes:

$$Q(\lambda, \lambda') = \sum_{q\in Q} \delta_0 log\, \pi_{q_0} P(O, q|\lambda') + \sum_{q\in Q}\left(\sum_{t=1}^{T} \delta_t log\, \pi_{q_{t-1}q_t}\right) P(O, q|\lambda')$$

$$+\sum_{q\in Q}\left(\sum_{t=1}^{T} \delta_t log\, b_{q_t}(o_t)\right) P(O, q|\lambda')$$

where $b_{q_t}(o_t) = \sum_{m=1}^{M} w_{q_t m}\, b_{q_t m}(o_t)$and

$$b_{jm}(o_t) \sim \mathcal{N}\left(o_t, \mu_{jm}, \Sigma_{jm}\right) = \frac{1}{|\Sigma_{jm}|^{\frac{1}{2}}(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}(o_t - \mu_{jm})^T \Sigma_{jm}^{-1}(o_t - \mu_{jm})}$$

It is straightforward that equations of the EM algorithm under the current set up become

$$\xi_t(i, j) = \frac{\delta_t \alpha_t(i) \alpha_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \delta_t \sum_{j=1}^{N} \alpha_t(i) \alpha_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \quad (1)$$

$$\gamma_t(i) = \frac{\delta_t \alpha_t(i) \beta_t(i)}{\sum_{i=1}^{T} \delta_t \alpha_t(i) \beta_t(i)} \quad (2)$$

$$\gamma_t(j, m) = \left[\frac{\delta_t \alpha_t(i) \beta_t(i)}{\sum_{i=1}^{T} \delta_t \alpha_t(i) \beta_t(i)}\right] \left[\frac{w_{jm} N(o_t, \mu_{jm}, \Sigma_{jm})}{\sum_{k=1}^{M} w_{jk} N(o_t, \mu_{jk}, \Sigma_{jk})}\right] \quad (3)$$

$$w_{jm} = \frac{\sum_{t=1}^{T} \delta_t \gamma_t(j,m)}{\sum_{t=1}^{T} \delta_t \sum_{k=1}^{M} \gamma_t(j,k)} \quad (4)$$

$$\mu_{jm} = \frac{\sum_{t=1}^{T} \delta_t \gamma_t(j,m) o_t}{\sum_{t=1}^{T} \delta_t \gamma_t(j,m)} \quad (5)$$

$$\Sigma_{jm} = \frac{\sum_{t=1}^{T} \delta_t \gamma_t(j,m)\left(o_t - \mu_{jm}^{new}\right)\left(o_t - \mu_{jm}^{new}\right)^T}{\sum_{t=1}^{T} \delta_t \gamma_t(j,m)} \quad (6)$$

$$\pi_i' = \gamma_1(i) \quad (7)$$

$$a_{ij}' = \frac{\sum_{t=1}^{T-1} \delta_t \xi_t(i,j)}{\sum_{t=1}^{T-1} \delta_t \gamma_t(i)} \quad (8)$$

EM Algorithm for Solving a GHMM for MWL

Iterations Counter: k: = 0

1. Set the initial values of the parameters of an GHMM
2. Calculation using the forward - backward algorithm of $\boldsymbol{\alpha_t(i)}$ and $\beta_t(i)$ variables

3. E-step: calculation of the variables $\boldsymbol{\xi_{\mathrm{t}}(i,j)}, \boldsymbol{\gamma_{\mathrm{t}}(i)}, \boldsymbol{\gamma_{t}(j,m)}$ using the relations (1), (2), (3) respectively
4. M-step: calculation of variables $\boldsymbol{\Sigma_{jm}}, \boldsymbol{\mu_{jm}}, \boldsymbol{w_{jm}}, \boldsymbol{\pi_{j}}, \boldsymbol{a_{ji}}$ using the relations (6), (5), (4), (7), (8)
5. If EM algorithm converges i.e. $\mathbf{l} - \mathbf{l}'<$ threshold then finish differently increase by 1 the iteration counter k: = k + 1 and repeat steps 2-5.

Appendix 4: k-MEANS Initialization Algorithm for GHMM

The first thing that must be done to train an HMM is to set the initial values of the parameters (λ). An easy approach is to use a random number generator process to select the values of probabilities taking into account constraints such as $\sum_{i=1}^{N} \pi_i = 1$. Since the EM algorithm guarantees convergence to a local maximum means that the random initial values lead to satisfactory results. However it would be ideal to find the total maximum if it exists, because then the HMM will provide the optimal fitting in the dataset. In addition to the literature [8] shows that good initial values especially in the case of continuous HMM lead to a substantial improvement of the results.

For a more efficient initialization process in [8] an adjusted to hmm K-means algorithm is described for the case of GHMM. Below the steps of the algorithm are briefly described for the case of hidden Markov models with normal distributions mixture:

- A large volume of data sequences are fed into the model. The parameters λ(π, A, Σ, μ, W) are initialized randomly. Then the likelihood of the sequence O is calculated based on the initialized HMM and the parameters λ.
- Based on the sequence data and the Viterbi algorithm produced the optimal state sequence of hidden states is determined. Then, based on this sequence of observations O is grouped into N groups based on the hidden state allocated to it through the Viterbi decoding process
- Each of the N groups obtained in step 2 is divided into M groups, equal to the number of component of Gaussian mixtures of initial HMM set up. The algorithm K-Means, applied in this step, is generally used for separating a dataset D in K disjoint subsets S_j, containing $D_j$points such as to minimize the function

$$\sum_{j=1}^{K}\sum_{d\in S_j}\left|x_d - \mu_j\right|^2$$

- where x_d is the price for d_{th}observation and μ_j is the average value for the data belonging to the subset S_j.
- The above steps have created N x M groups. For each group corresponding to a particular normal distribution of a Gaussian mixture corresponding to a particular hidden state new values of mean and variance are calculates using the conventional sample estimators of these variables. Additional weights in each normal mixture are calculated as: w_{ij}= number of data in the group corresponding to the member i of a mixture distribution j to all the data that are assigned to the mixture distribution j.
- In addition using the sequence of hidden states obtained in step two Table A of transitions probabilities is estimated as the number of transitions from state i to j against the number of occurrence of hidden state, and renewed. Finally the initial distribution π is updated as the total number of occurrences of hidden state I over the length of the sequence of hidden states.
- Then using the renewed parameters λthe likelihood of the observation sequence is recalculated O and if the difference with the original likelihood is smaller than a threshold d then the process stops. Otherwise the process is repeated from step 2.

References

1. *The Behavior of Stock Market Prices.* **Eugene Fama.** 1965, Journal of Business 38 (1): January 1965, pp. 34–105.

2. *A critique on efficient market hypothesis (EMH): Empirical evidence of return anomalies in 12 U.S. industry portfolios.* **Lee, Cheng Hsun George.** 2006, http://ir.lib.sfu.ca/retrieve/3840/etd2526.pdf.

3. *Stylized facts of daily return series and the hidden markov model.* **Ryd´en, T., Terasvirta, T. & Asbrink, S. .** 1998, Journal of Applied Econometrics 13(3), pp. 217–244.

4. *A tutorial on hidden Markov models and selected applications in speech recognition.* **Rabiner, Lawrence.** 1989, Proceedings of the IEEE 77, pp. 257-286.

5. *An inequality with applications to statistical estimation for probalistic functions of Markov processes and to a model for ecology.* **L. Baum and J. Eagon.** 1967, American Mathematical Society Bulletin, pp. 73:360-363.

6. *Maximum likelihood from incomplete data via the EM algorithm.* **A. Dempster, N. Laird, D. Rubin.** 1977, Journal of the Royal Statistical Society, pp. 1-38.

7. *A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models.* **Models, Bilmes, J. A.** 1998, International Computer Science Institute Berkeley California.

8. *Fundamentals of speech recognition.* **RABINER, Lawrence R. and JUANG, Biing-Hwang.** 1993, Prentice Hall, p. 100.

9. *Prediction of Financial Time Series with Hidden Markov Models.* **Yingjian Zhang.** 2004, Computing Science, Simon Fraser University.

10. *On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information measure.* **KRISHNAMURTHY, Vikram and MOORE, John B.** 1993, IEEE Transactions on Signal Processing 41.8, pp. 2557-2573.

11. *A survey of techniques for incremental learning of HMM parameters.* **KHREICH, Wael.** 2012, Information Sciences 197, pp. 105-130.

12. *Algorithmic Trading Hidden Markov Models on Foreign Exchange Data.* **Patrik Idvall, Conny Jonsson.** 2008, Department of Mathematics, Link¨opings Universitet.

13. *Hidden Markov models for time series: an introduction using R.* **Walter Zucchini, Iain L. MacDonald.** 2009, CRC Press.

14. *Stock market forecasting using hidden Markov model: a new approach.* **M.R. Hassan, B. Nath.** 2005, Proceedings of the Fifth International Conference on Intelligent Systems Design and Applications, pp. 192–196.

15. *A combination of HMM and Fuzzy model for Stock Market Forecasting.* **Md. Rafiul Hassan.** 2009, Neurocomputing Vol.72, pp. pages. 3439-3446.

16. *A fusion model of HMM ANN and GA for stock market forecasting.* **M.R. Hassan, B. Nath, M. Kirley.** 2007, Expert Systems with Applications 33 (1), pp. 171–180.

17. *Stock Market Analysis and Prediction using Hidden Markov Models.* **L, , L, and L, .** 2012, Application with code (Url: http://code.google.com/p/ftse/).

18. *A Hidden Markov Model Approach to Classify and Predict the Sign of Financial Local Trends.* **Manuele Bicego, Enrico Grosso and Edoardo Otranto.** 2008, Structural, Syntactic, And Statistical Pattern Recognition Lecture Notes in Computer Science, 2008, Volume 5342/2008, pp. 852-861.

19. *Hidden Markov models in computational biology: Applications to protein modeling.* **Krogh, A., Brown, M., Mian, I. S., Sjolander, K., and Haussler, D.** 1994, J. Mol. Biol. 235, pp. 1501-1531.

20. *Forecasting oil price trends using wavelets and hidden Markov models.* **SILVA, E, Souza, Edmundo de and LEGEY, Luiz FL.** 2010, Energy Economics 32.6, pp. 1507-1519.

21. *Hidden Markov Models for Speech Recognition.* **B. H. Juang; L. R. Rabiner.** 1991, Technometrics Vol. 33 No. 3., pp. 251-272.

22. *Vehicle Detection and Tracking in Car Video Based on Motion Model.* **Jazayeri, A. ; Cai, H. ; Zheng, J. Y. ; Tuceryan, M.** 2011, Intelligent Transportation Systems, IEEE Transactions, Issue:99,1.

23. *Performance of hidden Markov model and dynamic Bayesian network classifiers on handwritten Arabic word recognition.* **Jawad H. AlKhateeba, Olivier Pauplinb, Jinchang Renc, Jianmin Jiangb.** 2011, Knowledge-Based Systems Volume 24, Issue 5, July, pp. Pages 680-688.

24. *HMM based hand gesture recognition: A review on techniques and approaches.* **Moni, M.A., Ali, A.B.M.S.** 2009, Computer Science and Information Technology 2009. ICCSIT 2009. 2nd IEEE International Conference.

25. *Video-based face recognition using adaptive hidden Markov models.* **Xiaoming Liu, Tsuhan Cheng.** 2003, Proceedings. 2003 IEEE Computer Society Conference.

26. *Transient sonar signal classification using hidden Markov model and neural net.* **Kundu, A., Chen, G.C, Persons, C.E.** 1994, Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference, Issue Date: 19-22 Apr 1994, pp. On page(s): II/325 - II/328 vol.2.

27. *Automatic Detection of Abnormal Gait on Stairs.* **Jasper Snoe.** 2007, Computer Science University of Toront (Thesis) http://www.cs.toronto.edu/~jasper/thesis.pdf.

28. *Offline Signature Recognition using Hidden Markov Model.* **S. Adebayo Daramola, T. Samuel Ibiyemi.** 2010, International Journal of Computer Applications.

29. *Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models.* **España-Boquera, S., Castro-Bleda, M.J., Gorbe-Moya, J., Zamora-Martinez, F.** 2011, Pattern Analysis and Machine Intelligence IEEE Transactions.

30. *Heart signal recognition by Hidden Markov Models: the ECG case.* **Thoraval, Laurent, Guy Carrault, and Jean-Jacques Bellanger.** 1994, Methods of information in medicine 33.1, pp. 10-14.

31. *Analysis, classification, and coding of multielectrode spike trains with hidden Markov models.* **Radons, G., et al.** 1994, Biological cybernetics 71.4, pp. 359-373.

32. *A two-state Markov mixture model for a time series of epileptic seizure counts.* **Albert, Paul S.** 1991, Biometrics, pp. 1371-1381.

33. *A hidden Markov model approach to variation among sites in rate of evolution.* **Felsenstein, J., and Churchill.** 1996, Mol. Biol. Evol. 13 G.A., pp. 93–104.

34. *Recent applications of Hidden Markov Models in computational biology.* **Choo KH, Tong JC, Zhang L.** 2004, Genomics Proteomics Bioinformatics. 2004 May;2(2), pp. 84-96.

35. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* **Richard Durbin , Sean R. Eddy , Anders Krogh , Graeme Mitchison.** 1998, Book.

36. *The evolution and structure prediction of coiled coils across all genomes.* **Rackham OJ, Madera M, Armstrong CT, Vincent TL, Woolfson DN, Gough J.** 2010, Journal of Molecular Biology Volume 403, Issue 3, 29 October, pp. Pages 480-493.

37. *Modeling time series of animal behavior by means of a latent-state model with feedback.* **Zucchini, Walter, David Raubenheimer, and Iain L. MacDonald.** 2008, Biometrics 64.3, pp. 807-815.

38. *A time-series analysis of trends in firearm-related homicide and suicide.* **MacDonald.** 1994, International journal of epidemiology 23.1, pp. 66-72.

39. *Forecasting Turmoil in Indonesia: An Application of Hidden Markov Models.* **BOND, Joe.** 2004, nternational Studies Association Convention Montreal, pp. 17-21.

40. *A Family of Models for Drought.* **Zucchini, W.** 1991, Journal of Plants and Soil 27, pp. 1917-1923.

41. *Hidden Markov Models and Vector Quantization for Mobile Robot Localization.* **J. Savage, E. Marquez, F. Lepe-Casillas , and M.A. Morales A.** 2005, Proceeding (498) Robotics and Applications.

42. *Credit Card Fraud Detection Using Hidden Markov Model.* **SRIVASTAVA, Abhinav.** 2008, IEEE Transactions on Dependable Sec. Comput, pp. p.37-48.

43. *Traffic analysis attacks on Skype VoIP calls.* **Ye Zhua, Huirong Fub.** 2011, Computer Communications. Volume 34, Issue 10, 1 July, pp. Pages 1202-1212.

44. *Internet traffic modeling by means of Hidden Markov Models.* **Alberto Dainottia, Antonio Pescapéa, Pierluigi Salvo Rossib, Francesco Palmieric, Giorgio Ventrea.** 2008, Computer Networks Volume 52, Issue 14, pp. Pages 2645-2662.

45. *A Framework for Automatic Human Emotion Classification Using Emotion Profiles.* **Mower, L, and J, Narayanan M.** 2011, Audio Speech and Language Processing IEEE Transactions.

46. *Volcano-seismic Signal Detection and Classification Processing using Hidden Markov Models. Application to San Cristóbal Volcano, Nicaragua.* **Ligdamis A. Gutiérrez, Jesús Ibañéz, Guillermo Cortés, Javier Ramírez, M. Carmen Benítez, Virginia Tenorio, Isaac Álvarez.** 2009, IGARSS (4), pp. 522-525.

47. *Long swings in the dollar: Are they in the data and do market know it?* **Engel, C.,J. D. Hamilton.** 1990, American Economic Review 80, pp. 687–71.

48. *Can the Markov Switching Model Forecast Exchange Rates?* **Engel, C.** 1994, Journal of International Economics 36, pp. 151–165.

49. *Exchange Rates and Markov Switching Dynamics.* **CHEUNG, Yin-Wong and ERLANDSSON, Ulf G.** 2004, Journal of Business & Economic Statistics.

50. *Modelling exchange rates using regime switching model.* **Mohd Tahir, I & Zaidi Isa.** 2006, Journal Sains Malaysiana.35(2), pp. 55-62.

51. *Modelling returns on stock indices for western and central european stock exchanges - markov switching approach.* **Bialkowski J.** 2003, South-Eastern Europe Journal of Economics, pp. 81–100.

52. *Volatility Forecasts in Financial Time Series with HMM-GARCH Models.* **Xiong-Fei Zhuang,Lai-Wan Chan.** 2004, Intelligent data engineering and automated learning – ideal 2004. Lecture notes in computer science, 2004, volume 3177.

53. *Volatility Estimation via Hidden Markov Models.* **Alessandro ROSSI Giampiero M. GALLO.** 2005, Journal of Empirical Finance Volume 13, Issue 2, March 2006, pp. Pages 203-230.

54. *A hidden Markov regime-switching model for option valuation.* **Chuin Ching Liew, Tak Kuen Siu.** 2010, Insurance: Mathematics and Economics vol. 47 issue 3, pp. pages 374-384.

55. *Predicting trend in the next-day market by Hierarchical Hidden Markov Model.* **Troiano, L. Kriplani, P.** 2010, Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on Issue Date: 8-10 Oct., pp. On page(s): 199 – 204.

56. *Taking Time Seriously: Hidden Markov Experts Applied to Financial Engineering.* **Shi, S.,A. S. Weigend.** 1997, Proceedings of the IEEE/IAFE 1997 Conference on Computational Intelligence for Financial Engineering (CIFEr'97.

57. *A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle.* **Hamilton, J. D.** 1989, Econometrica March.

58. *Poisson Hidden Markov Models for Time Series of Overdispersed Insurance Counts.* **Paroli r., Redaelli, g., Spezia, l.** 2000, Proceedings of the XXXI International ASTIN Colloquium (Porto Cervo, 17-20 settembre 2000), Istituto Italiano degli Attuari, Roma, pp. 461-475.

59. *Bond Pricing in a Hidden Markov Model of the Short Rate.* **Land´en, C.** 2000, Finance and Stochastics 4, pp. 371-389.

60. *Filtering and forecasting commodity futures prices under an HMM framework.* **Tenyakov, Anton.** 2013, Energy Economics 40, pp. 1001-1013.

61. *Parameter estimation of an asset price model driven by a weak hidden Markov chain.* **Xiaojing Xia,Rogemar Mamon.** 2011, Economic Modelling Volume 28, Issues 1-2, January-March, pp. Pages 36-46.

62. *A hidden Markov model for space-time precipitation.* **Zucchini, W., Guttorp, P.** 1991, Water Resources Research 27(8).

63. *Modelling Portfolio Defaults Using Hidden Markov Models with Covariates.* **Konrad Banachewicz, Andre Lucas, Aad van der Vaart.** 2008, The Econometrics Journal, pp. 155-171.

64. *An Input Output HMM Architecture.* **Yoshua Bengio, Paolo Frasconi.** 1995, Advances in Neural Information Processing Systems, Vol. 7, pp. 427-434.

65. *Input-Output HMM 'I for Sequence Processing.* **Y. Bengio, P. Frasconi.** 1996, IEEE Transactions on Neural Networks, Vol. 7(5), pp. 1231-1249.

66. *Application of Hidden Markov Models and Hidden Semi-Markov Models to Financial Time Series.* **Jan Bulla.** 2006, Economics and Business Administration of the Georg-August-University of G¨ottingen.

67. *Hidden semi-Markov models.* **S Yu.** 2010, Artificial Intelligence, pp. 215—243.

68. *Visual Workflow Recognition Using a Variational Bayesian Treatment of Multistream Fused Hidden Markov Models.* **CHATZIS, Sotirios P. and KOSMOPOULOS, Dimitrios.** 2012, Circuits and Systems for Video Technology, IEEE Transactions on 22.7, pp. 1076-1086.

69. *Markov switching regimes in a monetary exchange rate model.* **Macdonald, R.** 2005, Economic Modelling 22 (3)., pp. 485-502.

70. *Stock market dynamics in a regime-switching asymmetric power GARCH model.* **Ane, T., and L. Ureche-Rangau.** 2006, International Review of Financial Analysis. 15, pp. 109-12.

71. *Regime-Switching Models.* **James D. Hamilton.** 2005, Palgrave Dictionary of Economics http://dss.ucsd.edu/~jhamilto/palgrav1.pdf.

72. *Finite Mixture and Markov Switching Models.* **Frühwirth-Schnatter, Sylvia.** 2006, Book.

73. *The Hierarchical Hidden Markov Model.* **S. Fine, Y. Singer and N. Tishby.** 1998, Analysis and Applications, Machine Learning, vol. 32, no. 1, pp. 41-62.

74. *Policy recognition in the abstract hidden Markov models.* **H. Bui , S. Venkatesh, G. West.** 2002, J. Artif. Intell. Res. vol. 17, p. p.451 .

75. *Modeling Individual and Group Actions in Meetings: A Two-Layer HMM Framework.* **Dong Zhang Gatica-Perez, D. Bengio, S. McCowan, I. Lathoud, G.** 2004, Computer Vision and Pattern

Recognition Workshop 2004. CVPRW '04. Conference Issue Date: 27-02 June 2004, pp. On page(s): 117 – 117.

76. *Recognition of Human Activities Using Layered Hidden Markov Models.* **Perdikis, S., D. Tzovaras, and M. G. Strintzis.** 2008, CIP Workshop.

77. *Interactive Hidden Markov Models and Their Applications.* **Ching, W., Fung, E., Ng, M., Siu, T. and Li, W. .** 2007, IMA Journal of Management Mathematics 18, pp. 85-97.

78. *Modeling Default Data Via an Interactive Hidden Markov Model.* **Wai- Ki Ching, Tak Kuen Siu, Li-min Li, Tang Li and Wai-Keung Li.** 2009, Computational Economics Volume 34, Number 1, pp. 1-19.

79. *HMMs and coupled HMMs for multi-channel EEG classification.* **Zhong, S., and Ghosh, J.** 2002, Proceeding IEEE Int. Joint Conference on Neural Network.

80. *Hidden Markov Mixtures of Experts for Prediction of Non-Stationary Dynamics.* **S. Liehr, K. Pawelzik, J. Kohlmorgen, S. Lemm and K.-R. Müller.** 1999, NNSP'99 Neural Networks for Signal Processing IX, pp. 195-204.

81. *Time-line hidden Markov experts for time series prediction.* **Wang, X., Whigham, P., Deng, D., and Purvis, M.** 2003, Proceedings of IEEE International Conference on Neural Networks and Signal Processing (ICNNSP'03), pp. 786-789.

82. *Profile hidden Markov models.* **Eddy SR .** 1998, Bioinformatics 1998;14(9), pp. 755-63.

83. *Inference in Hidden Markov Models.* **Cappé, O., and E. Moulines. T., Rydén.** 2009, Book.

84. *A second-order HMM for high-performance word and phoneme-based continuous speech recognition.* **J. Mari, D. Fohr, J. Junqua.** 1996, pp. 435—438.

85. *Automatic word recognition based on second-order hidden Markov models.* **J.-F. Mari, J.-P. Haton, A. Kriouile.** 1997, IEEE Trans. Speech Audio Process., pp. 22—25.

86. *Estimating the pen trajectories of static signatures using hidden Markov models.* **NEL, E.-M., DU PREEZ, Johan A. and HERBST, Ben M.** 2005, IEEE Trans. Pattern Anal. Mach. Intell., pp. 1733-1746.

87. *Data mining using hidden Markov models (HMM2) to detect heterogeneities into bacteria genomes.* **C. Eng, A. Thibessard, S. Hergalant, J.-F. Mari, P. Leblond.** 2005, Journées Ouvertes Biologie, Informatique et Mathématiques–JOBIM.

88. *Learning to automatically detect features for mobile robots using second-order hidden Markov models.* **O. Aycard, J.-F. Mari, R. Washington.** 2004, Int. J. Adv. Robotic Syst., pp. 231—245.

89. *Efficient backward decoding of high-order hidden Markov models.* **H.A. Engelbrecht, J.A. du Preez.** 2010, Pattern Recognition, pp. 99-112.

90. *Finite Mixture Models.* **G. McLachlan, D. Peel.** 2000, Wiley Series in Probability and Statistics.

91. *Margin-maximizing classification of sequential data with infinitely-long temporal dependencies.* **Sotirios P. Chatzis.** 2013, Expert Systems with Applications, pp. 4519—4527.

92. *On Prediction Using Variable Order Markov Models.* **Ron Begleiter, Ran El-Yaniv, Golan Yona.** 2004, Journal of Machine Learning Research, pp. 385-421.

93. *Bayesian Variable Order Markov Models.* **Christos Dimitrakakis.** 2010, pp. 161-168.

94. *A Nonstationary Hidden Markov Model with Approximately Infinitely-Long Time-Dependencies.* **Sotirios P. Chatzis, Dimitrios I. Kosmopoulos, George M. Papadourakis.** 2014, pp. 51-62.

95. *VOGUE: A Variable Order Hidden Markov Model with Duration based on Frequent Sequence Mining.* **M J. Zaki, C. D. Carothers, B. K. Szymanski.** 2010, ACM Transactions on Knowledge Discovery from Data, pp. 1-31.

96. *Improvements to the Sequence Memoizer.* **J. Gasthaus, Y. W. The.** 2011.

97. *EM procedures using mean field-like approximations for Markov model-based image segmentation.* **G. Celeux, F. Forbes, N. Peyrard.** 2003, Pattern Recognition, pp. 131-144.

98. *The mean field theory in EM procedures for Markov random fields.* **J. Zhang.** 1993, IEEE Transactions on Image Processing, pp. 27-40.

99. *Large margin Gaussian mixture modeling for phonetic classification and recognition.* **F. Sha, L. K. Saul.** 2006, Proceedings of ICASSP 2006, pp. 265—268.

100. *The Sequence Memoizer.* **F. Wood, J. Gasthaus, C. Archambeau, L. James, Y. W.** 2011, Communications of the ACM, pp. 91-98.

101. *Nonstationarities in stock returns.* **C at alin St aric a, Clive Granger.** 2005, Review of economics and statistics, pp. 503-522.

102. *Modelling financial time series.* **Stephen J. Taylor.** 2007, World Scientific Publishing Company.

103. *Empirical properties of asset returns: stylized facts and statistical issues.* **Rama Cont.** 2001, Quantitative Finance, pp. 223—236.

104. *A long memory property of stock market returns and a new model.* **Zhuanxin Ding, Clive WJ Granger, Robert F. Engle.** 1993, Journal of empirical finance, pp. 83-106.

105. *Volatility jumps.* **Viktor Todorov, George Tauchen.** 2011, Journal of Business & Economic Statistics, pp. 356-371.

106. *Do stock prices and volatility jump? Reconciling evidence from spot and option prices.* **Bjørn Eraker.** 2004, The Journal of Finance, pp. 1367-1404.

107. *The impact of jumps in volatility and returns.* **Bjørn Eraker, Michael Johannes, Nicholas Polson.** 2003, The Journal of Finance, pp. 1269-1300.

108. *Stylized facts of financial time series and hidden semi-Markov models.* **Jan Bulla, Ingo Bulla.** 2006, Computational Statistics & Data Analysis, pp. 2192-2209.

109. *Robust Sequential Data Modeling Using an Outlier Tolerant Hidden Markov Model.* **Sotirios P. Chatzis, Dimitrios I. Kosmopoulos, Theodora A. Varvarigou.** 2009, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1657-1669.

110. *A practical guide to volatility forecasting through calm and storm.* **C. T. Brownlees, R. F. Engle, B. T. Kelly.** 2009, Available at SSRN: http://ssrn.com/abstract=1502915.

111. *ECB fixing exchange rates.* **ECB.** ECB fixing exchange rates: https://www.ecb.europa.eu/stats/exchange/eurofxref/html/index.en.html.

112. *Copula processes.* **A. G. Wilson, Z. Ghahramani.** 2010, Advances in Neural Information Processing Systems.

113. *Benchmarks and software standards: A case study of GARCH procedures.* **B.D. McCullough, C.G. Renfro.** 1998, Journal of Economic and Social Measurement, pp. 59-71.

114. *Benchmarks and the accuracy of GARCH model estimation.* **C. Brooks, S.P. Burke, G. Persand.** 2001, International Journal of Forecasting, pp. 45-56.

115. *Gaussian Process-Mixture Conditional Heteroscedasticity.* **Emmanouil A. Platanios, Sotirios P. Chatzis.** 2014, IEEE Trans. Pattern Anal. Mach. Intell., pp. 888-900.

116. *Generalized autoregressive conditional heteroskedasticity.* **T. Bollerslev.** 1986, Journal of Econometrics, pp. 238—276.

117. *Autoregressive conditional heteroskedasticity models with estimation of variance of United Kingdom inflation.* **R. Engle.** 1982, Econometrica, pp. 987—1007.

118. *Mixed Normal Conditional Heteroskedasticity.* **M. Haas, S. Mittnik, M. Paolella.** 2004, Journal of Financial Econometrics, pp. 211-250.

119. *Variational Heteroscedastic Gaussian Process Regression.* **Miguel Lázaro-Gredilla, M. Tsitsias.** 2011.

120. *Some new stylized facts of floating exchange rates.* **James R. Lothian.** 1998, Journal of International Money and Finance, pp. 29-39.

121. *Forecasting crude oil market volatility: Further evidence using GARCH-class models.* **Yu Wei, Yudong Wang, Dengshi Huang.** 2010, Energy Economics, pp. 1477-1484.

122. *Modelling and forecasting volatility in the gold market.* **Stefan Trück, Kevin Liang.** 2012, International Journal of Banking and Finance, pp. 48-80.

123. *Statistical Learning Theory.* **V. N. Vapnik.** 1998, Wiley.

124. *Random Forests.* **L. Breiman.** 2001, Machine Learning, pp. 5-32.

125. *Modeling and Trading the EUR/USD Exchange Rate Using Machine Learning Techniques.* **Konstantinos Theofilatos, Spiros Likothanassis, Andreas Karathanasopoulos.** 2012, Engineering Technology & Applied Science Research, pp. 269-272.

126. *Forecasting the Taiwan Stock Market with a Novel Momentum-based Fuzzy Time-series.* **Tai-Liang Chen.** 2012, Review of Economics & Finance, pp. 38-50.

127. *A type-2 fuzzy time-series model for stock index forecasting.* **K. H. Huarng, Yu H. K.** 2005, Physica A, pp. 445-462.

128. *The application of neural networks to forecast fuzzy time series.* **K. H. Huarng, Yu T. H. K.** 2006, Physica A, pp. 481-491.

129. *Forecasting the Stock Market with Linguistic Rules Generated from the Minimize Entropy Principle and the Cumulative Probability Distribution Approaches.* **C.H. Su, T.L. Chen, C.H. Cheng, Y.C. Chen.** 2010, Entropy, pp. 2397-2417.

130. *International convergence of capital measurement and capital standards: A revised framework.* **Basel Committee on Banking Supervision.** 2005.

131. *Basel III: A global regulatory framework for more resilient banks and banking systems.* **Basel Committee on Banking Supervision.** 2010.

132. *Studies on the Validation of Internal Rating Systems.* **Basel Committee on Banking Supervision.** 2005.

133. *Financial ratios: Discriminant analysis and the prediction of corporate bankruptcy.* **E. I. Altman.** 1968, Journal of Finance, pp. 589—609.

134. *Forecasting US bond default ratings allowing for previous and initial state dependence in an ordered probit model.* **Paul Mizen, Serafeim Tsoukas.** 2012, International Journal of Forecasting, pp. 273-287.

135. *Consumer credit scoring: do situational circumstances matter?* **Robert B. Avery, Paul S. Calem, Glenn B. Canner.** 2004, Journal of Banking & Finance, pp. 835-856.

136. *Comparison of Credit Scoring Models on Probability of Default Estimation for US Banks.* **Gurný Petr, Martin Gurný.** 2013, Prague Economic Papers, pp. 163-181.

137. *Financial ratios and the probabilistic prediction of bankruptcy.* **James A. Ohlson.** 1980, Journal of accounting research, pp. 109-131.

138. *Combining bond rating forecasts using logit.* **Mark Kamstra, Peter Kennedy, Teck-Kin Suan.** 2001, Financial Review, pp. 75-96.

139. *An Exploratory Technique for Investigating Large Quantities of Categorical Data.* **Kass GV.** 1978, Applied Statistics, pp. 119—127.

140. *Classification and regression trees.* **Leo Breiman, Jerome Friedman, Charles J. Stone, R.A. Olshen.** 1984, CRC press.

141. *Bankruptcy prediction with industry effects.* **Sudheer Chava, Robert A. Jarrow.** 2004, Review of Finance, pp. 537-569.

142. *Forecasting bankruptcy more accurately: A simple hazard model.* **Tyler Shumway.** 2001, The Journal of Business, pp. 101-124.

143. *Regression Models and Life Tables (with Discussion).* **Cox DR.** 1972, Journal of Royal Statistical Society Series B, pp. 187—220.

144. *A time-dependent proportional hazards survival model for credit risk analysis.* **J.K. Im, D. W. Apley, C. Qi, X. Shan.** 2012, Journal of the Operational Research Society, pp. 306-321.

145. *Reduced form vs. structural models of credit risk: A case study of three models.* **Navneet Arora, Jeffrey R. Bohn, Fanlin Zhu.** 2005, Journal of Investment Management, pp. 43-45.

146. *On the pricing of corporate debt: The risk structure of interest rates.* **Robert C. Merton.** 1974, The Journal of Finance, pp. 449-470.

147. *The pricing of options and corporate liabilities.* **Fischer Black, Myron Scholes.** 1973, The journal of political economy, pp. 637-654.

148. *Pricing Derivatives on Financial Securities Subject to Credit Risk.* **Robert A. Jarrow, Stuart M. Turnbull.** 1995, Journal of Finance, pp. 53-85.

149. *Predicting bond ratings using neural networks: a comparison with logistic regression.* **John J. Maher, Tarun K. Sen.** 1997, Intelligent Systems in Accounting Finance and Management, pp. 59-72.

150. *Bankruptcy prediction using neural networks.* **Rick L. Wilson, Ramesh Sharda.** 1994, Decision support systems, pp. 545-557.

151. *A neural network model for bankruptcy prediction.* **Marcus D. Odom, Ramesh Sharda.** 1990.

152. *Application of support vector machines to corporate credit rating prediction.* **Young-Chan Lee.** 2007, Expert Systems with Applications, pp. 67-74.

153. *A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach.* **Kyoung-jae Kim, Hyunchul Ahn.** 2012, Computers & Operations Research, pp. 1800-1811.

154. *Integrating nonlinear graph based dimensionality reduction schemes with SVMs for credit rating forecasting.* **Shian-Chang Huang.** 2009, Expert Systems with Applications, pp. 7515-7518.

155. *A study of Taiwan's issuer credit rating systems using support vector machines.* **Wun-Hwa Chen, Jen-Ying Shih.** 2006, Expert Systems with Applications, pp. 427-435.

156. *Credit rating with a monotonicity-constrained support vector machine model.* **Chih-Chuan Chen, Sheng-Tun Li.** 2014, Expert Systems with Applications, pp. 7235-7247.

157. *A hybrid ensemble approach for enterprise credit risk assessment based on Support Vector Machine.* **Gang Wang, Jian Ma.** 2012, Expert Systems with Applications, pp. 5325-5331.

158. *Using Gaussian process based kernel classifiers for credit rating forecasting.* **Shian-Chang Huang.** 2011, Expert Systems with Applications, pp. 8607-8611.

159. *A hybrid KMV model random forests and rough set theory approach for credit rating.* **Ching-Chiang Yeh, Fengyi Lin, Chih-Yu Hsu.** 2012, Knowledge-Based Systems, pp. 166—172.

160. *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II.* **Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, T Meyarivan.** 2000, pp. 849-858.

161. *A hidden Markov chain model for the term structure of bond credit risk spreads.* **Lyn C. Thomas, David E. Allen, Nigel Morkel-Kingsbury.** 2002, International Review of Financial Analysis, pp. 311-329.

162. *Analysis of default data using hidden Markov models.* **Giampieri Giacomo, Mark Davis, Martin Crowder.** 2005, Quantitative Finance, pp. 27-34.

163. *Modeling default risk via a hidden Markov model of multiple sequences.* **Wai-Ki Ching, Ho-Yin Leung, Zhenyu Wu, Hao Jiang.** 2010, Frontiers Of Computer Science In China, pp. 187-195.

164. *A hidden Markov model of credit quality.* **RJ Elliott MW Korolkiewicz.** 2008, Journal of Economic Dynamics and Control, pp. 3807-3819.

165. *A Double HMM approach to Altman Z-scores and credit ratings.* **Robert J. Elliott, TakKuen Siu, Eric S. Fung.** 2014, Expert Systems with Applications, pp. 1553-1560.

166. *ML estimation of the t distribution using EM and its extensions ECM and ECME.* **C. Liu, D. Rubin.** 1995, Statistica Sinica, pp. 19-39.

167. *Robust mixture modeling using the t-distribution.* **D. PEEL and G. J. MCLACHLAN.** 2000, Statistics and Computing (2000) 10, pp. 339–348.

168. *Cross-Sectional Distributional Properties Of Financial Ratios In Belgian Manufacturing Industries: Aggregation Effects And Persistence Over Time.* **W. Buijink, M. Jegers.** 1986, Journal of Business Finance & Accounting, pp. 337-362.

169. *On the long-term stability and cross-country invariance of financial ratio patterns.* **Paavo Yli-Olli, Ilkka Virtanen.** 1989, European Journal of Operational Research, pp. 40-53.

170. *Some basic properties of financial ratios: Evidence from an emerging capital market.* **Zulkarnain Muhamad Sori, Mohamad Ali Abdul Hamid, Annuar Md Nassir, Shamsher Mohamad Sori.** 2006, International Research Journal of Finance and Economics, pp. 71-87.

171. *Some basic properties of accounting ratios.* **Geoffrey Whittington.** 1980, Journal of Business Finance & Accounting, pp. 219-232.

172. *Data Preprocessing for Supervised Leaning.* **S. B. Kotsiantis, D. Kanellopoulos, P. E. Pintelas.** 2006, International Journal Of Computer Science.

173. *Multi-interval discretization of continuous-valued attributes for classification learning.* **Irani, Keki B.** 1993.

174. *Pattern Recognition and Machine Learning.* **Christopher M. Bishop.** 2006, Springer.

175. *Variational Inference for Dirichlet Process Mixtures.* **David M. Blei, Michael I. Jordan.** 2006, Bayesian Analysis, pp. 121-144.

176. *Multivariate skew t-distribution.* **A.K Gupta.** 2003, Statistics, pp. 359-363.

177. *The multivariate skew-normal distribution.* **A. Azzalini, A Dalla Valle.** 1996, Biometrika, pp. 715-726.

178. *The Weighted Likelihood.* **Feifang Hu, James V. ZidekSource.** 2002, The Canadian Journal of Statistics Vol. 30 No. 3(Sep. 2002), pp. 347-371.

179. *A mixture model of two different distributions approach to the analysis of heterogeneous survival data.* **ERIŞOĞLU, Ülkü, ERIŞOĞLU, Murat and EROL, Hamza.** 2011, International Journal of Computational and Mathematical Sciences 5, pp. 75-79.

180. *A comparison of several time-series models for assessing the value at risk of shares.* **Zucchini, Walter, and Kristin Neumann.** 2001, Applied Stochastic Models in Business and Industry 17.1 , pp. 135-148.

181. *The Expectation Maximization Algorithm A short tutorial.* **Sean Borman.** 2004, http://www.seanborman.com/publications/EM_algorithm.pdf.

182. *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains.* **L. E. Baum, T. Petrie, G. Soules, and N. Weiss.** 1970, Ann. Math. Statist. vol. 41 no. 1, pp. 164-171.