Study of Anomaly detection in diverse populations using Probabilistic Graphical Models

Isaac Tarume

Supervisor: Benjamin Rosman

April 3, 2020

A thesis submitted to the Faculty of Science, University of the Witwatersrand, in fulfilment of the requirements for the degree of Master of Science.

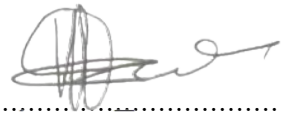
School of Computer Science and Applied Mathematics University of the Witwatersrand

Abstract

Most credit card transactions are characterised by randomly changing patterns of behavioural characteristics of the card users involved. The behavioural features tend to vary greatly due to the diversity inherent in most populations. In order to detect fraud and anomalies efficiently in these typical diverse populations using machine learning models, the models must be robust enough to capture the complex user behaviours in their variety. To address these challenges, we used base Hidden Markov Models (HMMs) and then a more richly expressive model called hierarchical Hidden Markov Model (HHMM) that can capture more dimensions and latent variables, which enables it to perform better. Furthermore, the model also reduces the learning times, which is very important when it comes to evaluating diverse population domain such as online credit card transactions as these impacts the response times. We evaluate the performance of HHMM in both individual and diverse population-based credit card transaction anomalies and labelled real-world timeseries data.

Declaration

I, Isaac Tarume, declare that this Thesis is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.



.....

Contents

List of Figures

List of Tables

1 INTRODUCTION

As the world continues to embrace new digital methods and channels of payments, use of credit cards has become a convenience and necessary part of financial life. Despite the ease of use, credit card payments have also brought many downsides, the major one being that they are susceptible to fraud. Fraud is a multi-billion-rand business. It costs many firms billions of Rands in losses and fines every year and it is increasing all the time [36]. Most of these costs end up being paid by customers in various forms like increased rates, higher premiums and reduced service offerings [58]. Fraud includes practices that directly or indirectly result in unnecessary expenses being incurred by the payer. Fraud can be defined as intentional deception used in order to get benefits [8]. There are various kinds of fraud including financial, telecommunication, credit card, network intrusion, scientific and insurance fraud. In this research we focus on the financial fraud especially those involving credit and debit cards. This choice was made in line with notable increases in card fraud as reported by the South African Banking Risk Information Centre (SABRIC) [29].

1.1 Problem

South Africa lost hundreds of millions of Rands to fraud committed using credit and debit cards in the year 2017. Up to R250 million worth of credit card fraud in 2017 resulted from transactions where the card was not present [7]. According to the SABRIC report, more than fifty percent of the bank card fraud loss happened when the card is not present [3]. Higher rates imposed by financial providers to counter fraud related losses are a burden on consumers. Finding a solution to this problem will not only address the crime but also alleviate the consumer burden and allow the money saved to be used for other developmental outcomes by citizens.

Fraud, especially card fraud, can be prevented and the payments industry has introduced a number of initiatives in this regard including more secure and improved Europay, Mastercard and Visa (EMV) chip. The EMV payment method is based upon a technical standard for smart payment cards and for payment terminals and automated teller machines for smart cards

that store their data on integrated circuits in addition to magnetic stripes, and Personal Identity Number (PIN). These enhanced security features have replaced the magstripe cards which were more susceptible to fraud. Despite these mechanisms card fraud is still prevalent and, in some cases, even increasing. One of the solutions that can be considered to this problem is to detect fraud before or just when the transaction is about to happen.

There are two main categories in which credit card fraud is committed. The first category happens with a physical card present. This type of fraud usually occurs when the original card is lost or stolen. Another instance of this type of fraud is when a clone of the card is made through card skimming and the new copy is used to purchase goods or services illegally. The second category is called virtual credit card fraud in which the perpetrator does not use the physical card but the details of the card to purchase goods or services especially online. These details could have either been stolen from a database with these details or from an image of the card. This fraud is also known as Card Not Present (CNP) [44].

Other types of card fraud include the following:

1. Not received issued card fraud (NRI) refers to the scenario where genuinely issued card are seized by imposters before they reach the authentic users and are used fraudulently;

2. Stolen Card fraud relates to transaction made from a legitimate card stolen from the user;

3. Lost Card fraud relates to transactions made from a legitimately issued card after the owner lost the card; and

4. Counterfeit card fraud is a form of physical card fraud committed using an illegitimately produced card encoded with information obtained from a magnetic strip of genuinely issued card.

In this research we are mostly going to focus on CNP fraud since we would like to use historical data which forms patterns that distinguishes one card user from another, Chan *et al.* [10].

CNP fraud occurs when a fraudster gets access to your card details especially, the card number. A fraudster is likely to make different kinds or

frequencies of transactions to the original user. If the bank could detect these, they could alert the account holder sooner.

In this instance, the card holder is typically unaware that their card is being used fraudulently and often remain so until they look at their statement or get a call from the bank about an attempted purchase.

It is important to note that a fraudster is likely to make different kinds or frequencies of transactions to the original user. If the bank could detect these, they could alert the account holder sooner. So, we need a way to detect these suspicious transactions and alert the user in time. Additionally, only a small percentage of the data is fraudulent making the dataset very skewed, Abdallah *et al.* [2] which leads to numerous problems such as disability of models to discover patterns in the minority class data. Additionally, skewed data sets impact the performance of classifiers that tend to be overwhelmed by the majority data sets and ignore the minority instances when learning. This misclassification leads to a lot of false negatives.

Furthermore, the card user behaviour varies significantly from one user population to another. This variation in behaviour results in diverse populations which makes the observed transaction data not easy to classify and even more difficult to detect which results in a lot of false positives. Accurate detection of the fraudulent transaction is not only important on the individual user transactions but also for the whole diverse population. Continuous monitoring of card holder expenditures, including the time, amount and geographical coordinates of each purchase, bring in important features which can make it possible to use machine learning models which can highlight with certain degree of accuracy whether a particular transaction is fraudulent. A model is trained on genuine transactions which calculate a range of possible probability value on how a transaction is genuine. For any new transaction a probability value is also calculated by the model. If the probability is higher than the set threshold by the card issuer then the alarm is sent to the card owner for further investigations.

In summary, the major problem we investigated in this thesis is anomaly detection in time-series datasets which are skewed and diverse that have latent stochastic variables. A diverse population of credit card users resulting in transaction data with multi-modes, which makes it difficult to detect the

fraud according to Kou *et al.* [27]. Skewed datasets also make it very difficult to pick anomalies as they are only a small percent (less than 5 percent of the data)

A common approach to solve this problem would be through studying time series data such as credit card user transaction data behaviour by monitoring expenditure data using specific machine learning models which are designed to analyse patterns. We studied temporal models, as they look at how data changes over time (in this case the purchase history). We also looked at ways in which different variables including latent ones interact and affect the spending patterns of credit card users. A natural framework encompassing latent stochastic variables that form sequential patterns is a probabilistic graphical model (PGM). Probabilistic Graphical Models (PGMs) are defined as a group of statistical models whose variables and relations can be represented graphically, Jordan *et al.* [22].

Machine learning, mainly graphical models, has given solutions in detecting anomalies in skewed and incomplete datasets as in Hollmen*et al.* [21]. Credit card transactions depict diverse data which makes graphical models better placed for detecting fraudulent transactions since graphical models are used represent and manipulate data in a structured way while modeling uncertainty according to Phua *et al.* [34]. The graphical models can also help visualise the complex structure and layers in the data which would help to provide better insights into the model properties. Identifying what methods to use and applying those methods are the most important tenets of this research. As such this research focuses on the identification and application of methods.

1.2 Extent of Problem

In most scenarios, a fraudster gathers user or consumer card details and pose as the consumer and goes on to buy or transact online illegally. In such cases the card holder is typically unaware that their card is being used fraudulently. Due to the diverse nature of the credit card transaction data sets, the bank might also not be able to pick up the fraud fast enough, until the user sees the bank card statement at the end of the month.

The goal of any PGM learning is such that one can be able to capture the

distribution from which the data is generated. Capturing the distribution of the data efficiently allows the model to fit any new data points and be able to distinguish them as part of the distribution or not. In our case the model should capture most of the usage features such as spending patterns and changes to those patterns which distinguishes a credit card user or a group of credit card users in the same population category/group. From the observed credit card transactions data, which is diverse, it is true that there will be several distributions inherent in one data set and that it is not always possible to have all representation of the behavioural characteristics of the individual or the group of individuals. The observed data would have a lot of overlapping between normal and anomalous data points, therefore a learned distribution will only give an approximation of the actual characteristics of the individual.

The purpose of PGMs learning is such that we can come up with the most comprehensive representation of the data which will allow the model not to misclassify an anomaly as a false positive resulting in potential fraudulent transaction being missed by the bank. Missing potential anomalous transactions might be costly to financial institutions and ultimately to the customer due to increase in fees and premiums. The best model depends on the goal of learning. Our goal in this thesis is to preserve the accuracy and performance as the population variability increases. In section 3.2, we will discuss population variability in detail. In brief, we used it in this thesis to refers to the diverse behavioural characteristics between credit cardholder‘s population groups which makes it difficult to distinguish between a fraudulent behaviour and a normal one.
Also, the structure of the diverse data set will be as follows:

1. The structure of the transactions is stochastic in nature as is the case with most credit card users where spending patterns are almost unpredictable based on current latent state.

2. The latent states are such that they have high level slowly changing states or in some cases, static states based on certain card user categories such as gender, mid-level states which changes occasionally and low levels states which are changing at a much faster pace than any levels above.

3. The transactions themselves are discrete or they can be categorised into

various discrete groups such as low, lower medium, upper medium, high spend categories.

Although PGMs, especially Hidden Markov Models (HMMs) have been used in detecting anomalies in credit card transactions, most of the literature refers to scenarios where a model is built from transactions of single card, Raj *et al.* [39,44]. But a more robust anomaly detection system must be able to detect anomalies not only from large populations but also from different types of population groups. This is quite important in credit card modeling where on any given day there are thousands of transactions going through the system.

A good anomaly detection system especially for card fraud detection should not only train and depend on transactions from the same user to classify incoming transactions as fraud or not, but it must also leverage from what has already been observed in other populations. In other systems, models per user are learned, stored and retrieved when a transaction from that user is observed and the new transaction is compared to the previously observed patterns from the same user. Experimental results from these methods show very good results in detecting similar observed patterns but does not show evidence of exploring anomaly detection in diverse populations, Singh *et al.* [11,13,40].

Our primary problem is to identify fraud in a population within a variety of population groups.

Most models will perform differently depending on the goals. Anomaly detection models especially in credit card fraud face difficulties in learning due to noise, skewness and some missing data points. In addition to these general challenges which most PGMs should be able to solve, we would like to focus more on variability of the population. Variability within a population brings more complexity to the time series data. It is for this reason we feel strongly that basic PGMs like HMMs (section 2.3.2) will not be efficient in these scenarios. We will instead turn to richer models in the form of Hierarchical HMMs which we believe are able to model anomaly detection in a diverse population, as experienced in real life scenarios of credit card modeling. This we believe, they are able to do whilst preserving the performance

and accuracy of the model.

In this thesis, we study the performance of HMM on single population group vs diverse population groups to determine effects on model accuracy as the population variability increases. We also study the improved HMM performance in the form of a Hierarchical HMM (HHMM) on a single population group and vs a diverse population to explore the effect of this improvement on the accuracy of the model.

In summary, the main difference between the approach in this thesis and most existing time series approaches designed for anomaly detection is that, the latter mostly considers one step and flat models which have one level of hidden states. An HHMM has various levels of states which describe input sequences at different levels of granularity and can be viewed being in form connected lattice: the nodes of the lattice representing states, the edges representing transitions between states, Skounakis *et al.* [41]. HMM can deal with the highlighted challenges of noise, missing data and skewness efficiently but tend to degrade in performance when the more complex varied populations are introduced. Our contribution is to an enhanced model based on HHMMs which we believe will be able not only to deal with missing data, noise and skewness, but also, a wide variety of populations whilst preserving the efficiency and accuracy of the probabilistic model.

1.3 Summary of the Document

The next chapter will look at the background of the PGMs and especially HMMs in relation to their use in anomaly detection in time series data. We will discuss their strengths and weaknesses and where they were applied successfully. We will look at their limitations which motivated this research. Additionally, we will also discuss work related to anomaly detection in a diverse population in particular fraud detection in time series data. In chapter 3 we outline the methodology. We will look at the HMM and HHMM models, their representation, inference, learning and how they can best be used for time series data anomaly detection. Chapter 4 will lay out the experimental work that will be used to detect anomalies in diverse populations, starting with the standard models to more diverse models to effectively cater for the different aspects of the data to correctly represent all the aspects of the data

structures. We will also discuss data simulations in this chapter. Chapter 5 will discuss the results and conclude this study.

2 BACKGROUND AND RELATED WORK

2.1 Introduction

In this chapter we discuss anomaly detection in sequential data and explore related work in anomaly detection in sequential data sets.

2.2 Models for Sequential Data Modeling

2.2.1 Graphical Models for Anomaly detection

When exploring anomaly detection in diverse sequential data sets, it is important to note that data variables may be hidden (latent) or visible (observed). Diverse data sets can be described as data sets belonging to different population categories which can be differentiated through some unique characteristic such as credit card usage behavioural patterns. We want to investigate the use of latent variables in these data sets.

Diverse data sets, particularly highly structured sequential data such as credit card transactions data, have a lot of observable variability which can be modelled using multi-level latent random variables. The observed variability in the data comes from usage behavioural patterns of various user population groups in credit card user populations. The specific usage patterns are governed by the latent variables inherent in the data sets.

Most PGMs, particularly HMMs that use latent variables, provide simple and intuitive interpretation of the make-up of the data sets. These PGM models have shown great success in learning and representation of sequence data sets, Zhou *et al.* [57]. This is mainly due to their ability to model latent variables and their complexity by expressing them in terms of graphs, in which relationships between the variables is also implicitly carried along, Maes *et al.* [28]. In our literature review we found little evidence that suggests that PGMs like HMMs have been tested on diverse populations. Since HMMs, which are single-level latent models, have been tested on single category and non-diverse populations, we would like to explore them on diverse populations. Diverse populations are more likely to have multi-level latent variables which increases complexity. These diverse data sets, we believe, will need richer models to capture the deeper structure and relations between multi-level latent variable where lower levels variables have faster

transitioning states than latent variables in higher levels which would have slow transitions, Fine *et al.* [16]. In this case, transitions refer to the changing states of the latent variables which in turn cause changes in the observed variables in the specific user population. Below we will look at how different scholars have investigated various methods to detect anomalies in credit card data.

2.2.2 Non-graphical methods

Various approaches have been used to detect anomalies in sequential data such as credit card transactions with the historical data being used as input to build a model of what a real transaction looks like and what an anomalous transaction looks like. For example, Ghosh *et al.* [20], used neural networks to detect anomalies in credit card transactions. The system will not be able to identify fraud in new user as the prerequisite is for the model to have trained on large volumes of data containing both anomalous and normal labelled data sets. The papers do not show evidence that in diverse user population groups the system performance does not degrade and that it is able to model data with latent variables at various levels efficiently. Maes et al in 2002 proved that Bayesian, networks, which is are PGMs, were more accurate in learning than comparative neural networks according to Maes *et al.* [28]. However, the evaluation process was slower in Bayesian networks than for neural network.

Other researchers have employed approaches using one class for labelled data only. Kim *et al.* [24], implemented a system in retail operations where anomaly detection is performed in steps by using association rules. The association rules are created by searching data for frequent if-then patterns and calculate the fraud density of the data points of the real transaction to identify the most important relationships, and then generate the weighted fraud score in the proposed scheme using associative rules. Murad and Pinkas [30] used a profiler, which is derived from a distance function and a clustering algorithm for probability distributions for the customer behaviour, to analyse calls, daily and overall levels of normal behaviour of each telecommunication account. Kokkinakki and Angelika [25] used similarity trees to detect legit-

imate behaviour of customers against outlier and segregation of clusters to segregate each legitimate user's credit card transaction. In all these methods, the experiments do not explore instances where the data sets were diverse, in most cases the tested data sets were single category, non-diverse populations. Additionally, none of these models were adapted to multi-level latent variable models to explore diverse populations.

All these methods are useful in detecting anomalies where there is a lot of data available for training and when the type of anomalies being detected have been seen before by the model. There is however no evidence that their performance improves or remains the same when there is more population diversity. Furthermore, we also did not find evidence in literature to support that these non-probabilistic approaches can effectively handle data with latent variables and missing data points, Phua *et al.* [24, 25, 30, 34] Most researchers used probabilistic graphical models to identify anomalies in structures with latent variables and missing data. HMMs have been employed to detect anomalies in credit card transactions in a single-category non-diverse populations, Hollmen and Jaakko [21]. In the next section we look at HMMs use in anomaly detection.

2.3 HMMs Background

A Hidden Markov Model is a mathematical model with of two or more random variables in which the system being modeled is assumed to be a Markov process. A Markov process is a system in which a current occurrence of an event is only dependent on the state obtained in the previous event. The stochastic process will have one or more variables with one being hidden and not directly observable but will influence the one which is observable. An HMM can be represented in form of a simple dynamic Bayesian network. A Bayesian model, is a probabilistic graphical model that depicts, conditional dependencies between random variable sets, via a directed acyclic graph.

2.3.1 Discrete Markov Chain

A discrete Markov chain describes a sequence of random variables $S_1, S_2, S_3, ...$ with the Markov property, which says that the probability of moving to each

of the future states depends only on the present state and not on the past states, Tauchen and George [48]. Consider a system which can be described at any given time in N distinctive states S_1, S_2, $S_3...S_N$. Si is an individual state. The state at time instant t is denoted by St. At regularly spaced discrete times the system can undergo a change from one state to the other according to the set of probabilities associated with the transition from one state to another. If we can denote the changing time states as t_1, t_2, t_3... The general probability of this system can be depicted by the following

$$P(S_{t+1} = s \mid S_1 = s_1, S_2 = s_2...S_t = s_t) = P(S_{t+1} = s \mid S_t = S_t), \quad (1)$$

Equation (1) is known as the Markov assumption, where the general Markov property is assumed to hold. A stochastic process has the Markov property if the conditional probability distribution of next state states of the process depends only upon the present state, not on the sequence of events that preceded it.

The possible values of S_n form a countable set called the state space of the chain. Markov chains are often described by a sequence of directed graphs, where the edges of graph t are labelled by the probabilities of going from one state at time t to the other states at time $t + 1$,
$P(S_{t+1} = s \mid S_1 = s_1, S_2 = s_2, ..., S_t = s_t) = P(S_{t+1} = s \mid S_t = s_t)$
Given the right hand is independent of time thereby leading to the set of state transition probabilities a_{ij}, a transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j. State transition probabilities are the probability matrix distributions that are associated with transitions from one hidden state to another.

$$a_{ij} = P[S_t = S_j | S_{t-1} = S_i], 1 <= i, j <= N \quad (2)$$

State transition probabilities satisfy the following properties

$$a_{ij} >= 0, \quad (3)$$

each transition probability is non-negative. Zero probability would mean there is no transition between state i and j.

$$\sum a_{ij} = 1, \quad (4)$$

The sum of all the probabilities should add up to 1

This is an observable Markov model since the output of the process is the set of states at each instant of time where each state corresponds to a physical observation [38].

Bhusari *et al.* [5], demonstrated the use of discrete Markov chains with unobserved states to detect anomalies in non-diverse credit card data. The model had two sets of variables, one set of variables are hidden since their states are not directly observable. These variables were categories of items which are being purchased (such as luxuries, bills, services, tickets) that are not directly observable to the credit card authorising company. What is observed are the transaction amounts of the items being purchased. Therefore, the different categories of items and services being paid for can represent the hidden states, and the actual amounts can be modelled as the observable states in HMMs.

In a Discrete Markov Chain, particularly HHMs, the challenge of adjusting the parameters to account for observed states is solved by using the method for estimating parameters known as the Baum-Welch algorithm. The method for estimating the sequence of most likely states will solve the issue of determining the best sequence by using the backward-forward algorithm [4]. Alternatively, one could use the Viterbi algorithm to estimate the single most likely sequence of states.

2.3.2 Hidden Markov Model

HMMs can be presented as the simplest Dynamic Bayesian Network with both observed and latent variables. A Bayesian network is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (Figure 1). In Figure 1, v (dark-shaded circles) represent the observed variables, while the s (transparent circles) represent the hidden variables. The figure shows the evolution of a hidden Markov process from time 0 to T where for each hidden state transition, for example from state s_0 to state s_1 there is an emission of a symbol v_1. HMMs as probabilistic graphical models have been used for anomaly detection, chiefly fraud detection in credit card

transactions as demonstrated by Bhusari *et al.* [5]. The different categories of purchased items were mapped to hidden variables (s_i) and the different transaction amount ranges between low, medium and high, were mapped to the model as observed variables (v_i).

As highlighted earlier in this chapter, according to Rabiner *et al.* [38] an

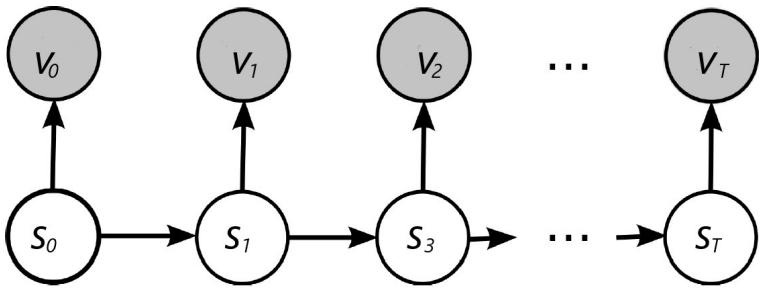


Figure 1: Hidden Markov Model [5]

HMM should have the following elements:

1. The number of states of the model, N. We denote the set of states $S = S_i$, where $i = 1; 2; ...; N$, represent states and Si, is a single state, St denotes state at time t.

2. M denotes total number of symbols which can be observed. For continuous observations, M is infinite. The set of symbols is denoted by $V = V_1; V_2; ...V_M$ where Vi , is a single symbol for a finite value of M.

3. The state transition probability matrix, which is a list of transition probabilities from one hidden state to another, is defined as
   $a = [a_{ij}]$,
   $$a_{ij} = P[S_t = S_j | S_{t-1} = S_i], 1 <= i, j <= N \tag{5}$$
   , where any state j can be reached from any other state i in a single step, we have
   $$a_{ij} >= 0 \tag{6}$$
   for all i, j. And
   $$\sum_{j=1}^{N} a_{ij} = 1, 1 <= i <= N. \tag{7}$$

4. The emission probability in state j is the probability of transitioning to a certain given observed state k based on hidden state j, is defined as $B = [b_j(k)]$, where

$$b_j(k) = P[V_k|q_t = S_j]1 <= k <= M, \tag{8}$$

and

$$\sum_{k=1}^{M} b_j(k) = 1, 1 <= j <= N. \tag{9}$$

5. The initial probability distribution over states. π specify the starting state i. It is defined by
$\pi = [\pi_i]$

$$\pi_i = P[S_1 = S_i], 1 <= i <= N \tag{10}$$

and

$$\sum_{i=1}^{N} \pi_i = 1 \tag{11}$$

6. HMM assume conditional independence on current observation to previous ones. The observation sequence V, can be represented as V = V_1, V_2, V_3,..... V_R, where V_t denotes one of the observation symbols V, from time t>=2. R represents the length of the observation sequence.

HMMs make two main assumptions:

1. the observation at some time t was generated by some process whose state S_t is hidden,

2. given the value of the previous state S_{t-1} the current hidden state S_t is independent of all the other states prior to S_{t-1} [17].

In any HMM, N represents the number of states in a model, M represents the permissible observation symbols in any state , A is the state transition matrix, B is the emission symbol matrix, initial state vector π. The notation λ = (A, B, π) is used to denote an HMM's full tuple including parameters with discrete probability distributions. This notation also includes N and M implicitly.

Various combinations of state sequences can produce an observation sequence V. For instance, a sequence, S=S_1, S_2,...S_R , where S_1, is the initial state. We can compute the probability that V was produced by this state as follows:

$$P(V|S,\lambda) = \prod_{t=1}^{R} P(V_t|S_t,\lambda), \tag{12}$$

Equation (12) can be expanded as

$$P(V|S,\lambda) = bS_1(V_1).bS_2(V_2)...bS_R(V_R), \tag{13}$$

We compute the probability of a specific sequence S as follows:

$$P(S|\lambda) = \pi_{S_1} a_{S_1S_2} a_{S_2S_3}...a_{S_{R-1}S_R} \tag{14}$$

, since the HMM property states that the current symbol is only affected by the previous observation only.

The joint probability of V and Q i.e. the probability that set of V observation sequence and set of S state sequence occur simultaneously is therefore the product of the two terms,

$$P(V,S|\lambda) = P(V|S,\lambda)P(S,\lambda) \tag{15}$$

Thus, the probability of producing V observation sequence, by a model denoted by λ is computed by adding the joint probabilities of the various viable state sequences S as follows:

$$P(V|\lambda) = \sum_{allS} P(V|S,\lambda)P(S|\lambda) \tag{16}$$

$$= \sum_{S_1,S_2,....,S_T} \pi_{S_1} b_{S_1}(V_1) a_{S_1S_2} b_{S_2}(V_2)....aS_{T-1}S_T b_(S_T) \tag{17}$$

We elaborate the equations above as follows:
At time ($t = 1$), the system is in state S_1 with probability π_{S_1}, and produce observation V_1. The process time then moves to $t + 1(t = 2)$ from t and the system transitions from make a S_1 to S_2 with probability $a_{S_1S_2}$, and produce observation V_2 with probability $b_{S_2}(V_2)$. The process proceeds until transition at time T from state S_{T-1} to state S_T with probability $a_{S_{T-1}S_T}$ and produce observation V_T with probability $b_{S_T}(O_T)$. Deriving the value of

$P(V|\lambda)$ straight from its definition (17) is mathematically rigorous, involving order of $2N_T^2$ calculations, since at every $t = 1, 2, ...T$, there are N possible states which can be reached, and for each such state there should be 2T for each term in the sum of (17). Computationally, its almost operationally infeasible to compute this probability, especially for large numbers of N and T. A procedure, forward–backward algorithm, is used to calculate $P(V|\lambda)$.

Before we discuss the forward-backward algorithm, we highlight three basic challenges which we have to solve for. The three basic challenges which must be solved when using HMMs are:

1. Given the observation sequence V=$V_1, V_2, V_3, ..V_T$ and model λ=(A, B, π) How do we efficiently compute $P(V|\lambda)$, the probability of the observation sequence given the model?

2. With observation symbols $V_1, V_2, V_3, ...V_T$ and with model λ. How to come up with an equivalent state sequence S=$S_1, S_2, S_3, ...S_T$ that corresponds to the observations sequence?

3. How do we fine-tune the model parameters λ(A,B,π) to optimise $P(V|\lambda)$?

Solving these challenges, one would employ dynamic programming paradigms (i.e. methods for solving a complex problem by breaking it down into a collection of simpler sub problems, solving each of those sub problems just once, and storing their solutions - ideally, using a memory-based data structure).

1. Solution to issue 1, computing probability of hidden sequence, we use the forward-backward algorithm. The algorithm involves two steps as described by Rabiner *et al.* [38];

   (a) Computing forward probabilities by summing all possible observation probabilities up to the final observation, If $\alpha_t(i)$ denotes observation sequence probability, from all possible state sequences up to the $i - th$ state.

   $$\alpha_t(i) = P(V_1, V_2, ...V_t, Q_t = S_i|\lambda) \tag{18}$$

   Then the independent likelihood of the observed sequence will be the sum of $\alpha_t(i)$ over all N states. The Forward algorithm which is a recursive algorithm has been used to calculate $\alpha_t(i)$ for the observation sequence of increasing length t, Rabiner *et al.* [37].

(b) Computing the backward probabilities by summing all possible observation probabilities which start with a certain state. Symmetrically we can have $\beta_t(i)$, the conditional probability of observation sequence from $V_{(t+1)}$ to the end, to be produced by all state sequences that start at the i-th state.

$$\beta t(i) = P(V_{T+1}, V_{T+2}, ...V_T, S_t = S_i|\lambda) \tag{19}$$

The probability of the partial observation sequence from t+1 to the end given state S_i at time t and the model λ The Backward Algorithm calculates recursively the posterior marginals of all hidden variables going backward along the observation sequence. The forward−backward algorithm can be used to find the most likely state for any point in time. It cannot, however, be used to find the most likely sequence of states. It is possible to get states at two times (t and t+1) that are both most likely at those time points but which have very little probability of occurring together. Viterbi algorithm is used for finding the most likely sequence.

2. Solution to issue 2, infer the most probable latent states path. We use the Viterbi algorithm has typically been used for this . The procedure identifies optimal latent states path that maximises the probability of the observed sequence, as proposed by Rabiner *et al.* [38]. If $\sigma_t(i)$ is the highest likelihood of the state path of length t, with i as the last state that emits the t first observations for the given model. Then we have:

$$\sigma_t(i) = maxP(S_1, S_2, ..., S_{t-1}; V_1, V_2, ..., V_t|S_t = S_i) \tag{20}$$

Viterbi uses the dynamic programming algorithm with same procedures as the forward–backward algorithm but with two deviations:

(a) It calculates the maximum likelihood in place of summation after every recursion cycle.

(b) It creates N by T matrix ψ to store inputs that maximise $\sigma_t(i)$ for each t and i. At backtracking step, the matrix is used to reveal the optimal state path.

3. Solution to issue 3, learning parameters A, B, π. We use the Baum-Welch algorithm to try and maximise the probability of the observation sequence according to the model λ.

   Expectation maximization can be summarised into the following steps according to [38]

   (a) Initialise the model λ, parameters (A, B, π)

   (b) Estimate $A_{it}, B_j(k)$ in the training data

   (c) Update the model parameters (A, B, π according to $A_{ij}, B_j(k)$

   (d) Repeat (b) and (c) till it converges

2.4 HMMs Related Work

The use of HMMs to detect anomalies in credit card transactions in a single-category non-diverse populations has already been explored Hollmen *et al.* [21]. PGMs, mainly HMMs, have widely been used when there is need for simplifying structures and relations between latent variables. They have been used in speech recognition and medical diagnosis successfully Wang *et al.* [51, 53]. In the area of anomaly detection (such as fraud detection), they have also been used with good results especially in handling latent variable in sequential data Raj *et al.* [39, 40, 55]. Furthermore, HMMs perform very well in learning data with complex structure and missing data points Cooke *et al.* [12]. This is due the use of a combination of probability distributions to denote conditional probabilities that exist between variables and also the use of graphs and nodes to illustrate the relationships between the variables Smyth *et al.* [18, 42].

This brings us to an important aspect of data observability in PGM learning. Data being modeled is not always complete and fully observable. This is so, for a number of reasons, one of them being that its virtually impossible to observe all the data for any given process, for instance from credit card fraud modeling usage patterns where users use cash to purchase goods, will not be observed. As a result, data is mostly likely incomplete, partially observable and it will contain hidden variables whose values are never observed in any of the training instances. This arises in the case where set of variables X is unknown, or they are known but were never observed due to various reasons such as in credit card transactions the merchant may only sends the total

purchase amount to credit card issuing banks and not the individual items and their categories of purchased goods, Murphy *et al.* [19, 31].

During training, HMMs, use the forward–backward algorithm to estimate the probabilities of the hidden states and their transitions. They also use the same algorithms to calculate the emission probabilities which means they can deal with both missing data and hidden variables. As shown in their paper Srivastava *et al.* The system they proposed was able to detect anomalies which represented fraud from as little as 20 transactions from a single user with an accuracy of 80 percent over various input data. In the same paper they only used the actual transaction amount as the only observable variable. Bui *et al.* [9] was able to improve the accuracy by using state hierarchy HMMs to detect anomalies from a small population of training data.

Wang *et al.* [53] proposed an even better solution in intrusion detection where smaller sequence of a program block was broken down into small streams called calls. An HMM model was then used for anomaly detections at each system call rather than sequences. In the same paper Wang *et al.* [53] proposes to make use of a sequence of system calls in a trace as observables. This method had results indicating better performance from smaller sequences compared to large ones. Although HMMs were used successfully in these cases to improve accuracy, population diversity and its effect on accuracy was not highlighted in these papers, hence our focus on it.

In 2004 Foster *et al.* [15] highlighted that most of anomalous data, particularly financial fraud data, contains less than 10 percent of fraudulent transactions, which makes most credit card transactions imbalanced data sets. Imbalanced data sets are difficult to train due to the overlapping in behavioural usage patterns of normal and fraudulent users. Since anomaly detection, mostly fraud detection, depend on observing previous behaviours of individuals and determining outliers from historical transactions, it also follows that due to the complexity of human behaviour and the diversity of transactions themselves, any normal transaction has equal chances of being legitimate or anomalous, Kavitha *et al.* [23]. A good system should be able to deal with population variability.

2.5 Challenges to the approaches

In this research we focused on PGMs, like HMMs, which can model latent variable. In the next section, we will look at the simple Hidden Markov Model before exploring the possibility of extending it into a hierarchical model by adding certain key elements. This thesis will argue that unlike the simple Hidden Markov Model, a hierarchical model will produce a more superior representation of the real world being modeled under the assumption of diverse populations.

2.6 Motivation for PGMs

Hierarchical models were originally developed for many natural sequences particularly handwriting and speech recognition as explained in and put forward by Fine *et al.* [16]. Their application to diverse populations, is novel and our contribution also includes various extension which seek to suit our circumstances. Our primary motivation is to be able to model the different random behavioural characteristics that are present in varied populations of credit card users.

2.7 Conclusion

In this chapter we have looked, in depth, at the HMM and its application to anomaly detection particularly fraud detection. We have discussed how we can learn the model parameters with the training data set and then use the trained model to compute a confidence score for next transactions to determine whether they can be suspected to be fraudulent or not. In the next chapter we will describe the methodology for exploring diverse cardholder populations. We will explore on the methodology to add some modifications to the standard HMM to be able to capture the peculiar characteristics of anomalous transactions. We believe that by modifying the structural representation and expanding the various latent states into hierarchies, we can improve the behaviour of HMM and therefore, solve the problem of diversity in credit card transactions.

3 METHODOLOGY

3.1 Introduction

This chapter describes our approach to anomaly detection in a diverse population for credit card transactions. Firstly, the chapter highlights the diverse population features and how they might impact performance. It then describes the methodology for modeling diverse credit card anomalies using a standard baseline HMM initially on single credit cardholder transactions (3.2.1 and 3.2.2) as proposed by Srivastava *et al.* [45] and then a second methodology on credit transactions from various categories of diverse cardholder population (3.2.3). In section 3.2.4 the effects of diversity on HMM modeling are explained including some insights that we believe are the reasons for the seemingly poor performance of flat models in categorisation. Thereafter the chapter is structured as follows: section3.3 tries to introduce hierarchical structures to the flat HMM to enable the model to model diverse populations much more effectively. In section 3.4 enhancements to the HMM to give the hierarchical nature is further explained and finally, HHMM learning and inference are clarified in section 3.6. In section 3.7, the chapter is concluded.

3.2 Diversity

In this thesis, population diversity is used to refer to the splitting and merging of behavioural characteristics of different cardholders across various population categories. It also refers to the extent to which these behavioural characteristics differ from each other. This diversity is due to the statistical variability in credit card users‘ population whereas individual behavioural characteristics data points around spending vary from one another. Statistical variations in user features is inherent naturally in human beings. This mixture of behavioural features makes normal credit card transactions datasets overlap with anomalous credit card datasets as seen in the time series examples in Figure 2. The figure shows that normal behaviour for one user is abnormal for another.

Part a) of Figure 2 shows normal user 1 and 2 transactions sequence and the same user transactions with anomalous transactions inside the sequence. Part b) shows normal user 3 and 4 transactions sequence and the same se-

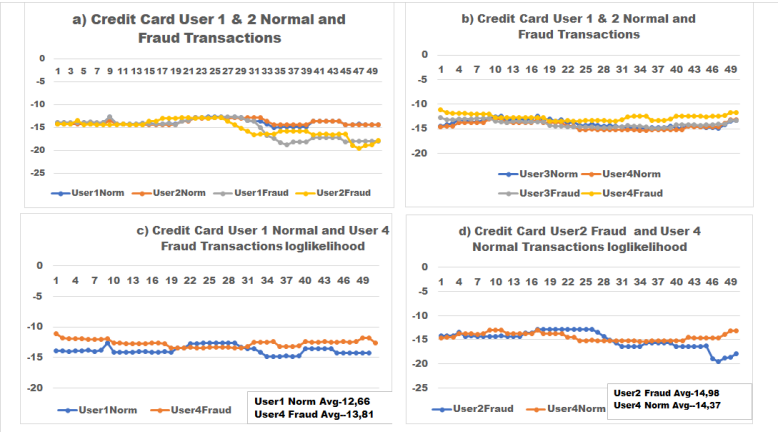


Figure 2: Statistical Variation in diverse Credit Card Transactional data, showing that normal behaviour for one user is abnormal for another.

quence with some anomalous transactions. However, Part c) shows normal transactions sequence of user 1 and anomalous transaction sequence of user 4 layered on top of each other. The average mean for the likelihood of the normal user 1 transactional sequence is 12.66 and the one for anomalous transactional sequence for user 4 is 13.81 respectively. The likelihoods are calculated by the forward–backward algorithm as explained in section 2.3.2, HMM Model equation 18,19. Part d) of the graph also shows user 2 transactions sequence which include anomalous transactions. User 4 normal transactions sequence are also shown in the same graph in Part d). The average likelihood of the sequences is 14.98 for user 2 anomalous and 14.37 for user 4 normal sequence respectively. The sequence likelihoods clearly overlap each other on different sections of the trajectory which might make it difficult for any classification method to separate them. The overlapping datasets create a challenge to the learning model as it becomes difficult to separate anomalous and normal transactions. Perhaps in such situations keeping individual models per user will be more appropriate. However, in cases where the historical transactions of the user do not exist, there will be a need to depend on the population trained model.

The challenge is due to certain behavioural characteristics that appear normal in one population category but only to be classified as anomalies in another category. This overlap in behaviour results in high false positive rates. In

anomaly detection (such as credit card fraud detection), it is important to reduce false positives, as false alarms are costly to the credit card issuing institutions. This cost may arise from the institution mobilising its resources to investigate the alarms. Additionally, sending false alarms is generally inconveniencing customers.

We will demonstrate three distinct models:

1. HMM Model per Credit Cardholder (Single user population)
2. HMM Model for All Credit Cardholders (Single category population)
3. HMM Model per Category of Credit Cardholders (Multiple category populations)

To explore the above models, we simulated three data sets as follows:

1. Category A population consists of simulated data sequences belonging to the same type of population. This means the credit card user usage patterns are similar in this group.
2. Category B populations consists of simulated data sequences belonging to the same type of population, however the behavioural usage patterns are different from Category A.
3. Category AB populations consists of equal mixture of data sequences from Category A and B.

Consider four Users: U1, U2, U3 and U4. U1 and U3 are high spenders who will buy anything and everything that they come across while U2 and U4 are conservatives who spend some time weighing the pros and cons before they buy something.

The HMM approaches in section 2.4 assumes all users have same behavioural usage patterns and therefore belong to the same population group. Some credit card user behavioural patterns are more like certain users than others due to various reasons. For instance, a conservative credit card user might not purchase as many items from a category and as frequently as another user. Another user, a high spender, might be purchasing items in all categories, such as groceries, clothes, tickets, and online merchandise at a much

higher frequency. If the assumption is that these users belong to the same population category, the downside of such an assumption is that, since the users might belong to different categories (for example Category A and B), the resulting transactional data from the user purchases will probably be diverse in nature. Our assumption is that this diversity might bring multi-level latent variables which are hierarchical in nature which will need to be considered when modeling the data, failure of which will result in the model performing poorly.

The poor performance could be as a result of a few challenges;

1. How to summarise the cardholder characteristics in the population? The data set in the third model above (HMM Model per Category of Credit Cardholders - Multiple category populations) will more likely be multimodal due to the diversity and therefore have various modes. The multimode might also mean that there are multi-level latent variables controlling the observed patterns in the diverse data sets. Effective identification of the modes and representation of the multi-level latent variables, we believe, will help in improving the performance of the model.

2. How to decide the subset user characteristic associated with a specific cardholder? Having identified that there are multiple modes, the second challenge will be to assign a new incoming sequence to the right population group. The model will need to check the incoming sequence distribution and match it to one of the diverse population categories and then predict any deviation from the learned distribution. This is particularly a difficult task due to data overlap which results in misclassification as discussed in section 3.2.

Most HMM approaches solved misclassification by clustering users into spending profiles, low, medium and high, Singh *et al.* [13, 14, 40]. But in these papers the population was non-diverse single category populations. There is also no evidence in the papers to suggest that these HMM approaches were effectively explored in diverse populations with multi-modes.

One way to solve the first challenge could be to consider defining a user profile for each user and use the profile to detect anomalies for that user. This is what the first model (Single HMM Model per Credit Cardholder)

attempts to use. Nonetheless, two problems arise from training single user profiles:

1. pre-defined single user population might not always be available in real life due to missing data as a result of some unobserved data points as in the case of a new user discussed in section 2.4 ; and

2. training a model only on previously observed patterns from one single user might mean that new anomalous patterns from the same user group will not be identified right up front. Inversely, normal behaviour will be incorrectly classified as anomalies resulting in costs.

3.2.1 HMM Model per Credit Cardholder

Every credit card holder has a unique spending profile and the model uses the deviation in this spending pattern on the purchasing amount of the latest 10 transaction sequence to identify any anomalies against a new transaction. If there is a deviation the system will reject the transaction and issue a warning message to the user.

We are going to look at various approaches to modeling anomalies in credit card transactions. One method of looking into modeling anomalies in credit card transactions is by considering a model in which, for each individual cardholder, an HMM is trained and maintained. The HMM model is trained on the user‘s historical purchases only. It is highly likely that if there are enough historic transactions from which the model can build a user profile, the model can detect any abnormal behaviour promptly. The approach is as per the below steps:

1. The HMM is fed with a sequence of emission symbols where a symbol represents a purchase. For purposes of demonstrating this method, this research only considers four symbols for the four spending profiles: low, lower medium, upper medium and high spending profile. Only one of the four symbols is observed at any given time when the system‘s hidden state changes.

2. The Baum Welch algorithm as explained in section 2.3.2, is used to learn the HMM parameters of each card holder. The algorithm initialises at uniform probabilities of π, A, B, and converges to a local maximum as per the domain function. The initial uniform probability for A means if there are N states, then the initial state probability for each state will be $1/N$. The initial state transition will also be uniform, 1/4 in our case since there are four emission symbols.

3. Subsequent to setting up the initial probabilities, training commences as follows:

   (a) initialisation of HMM parameters

   (b) forward algorithm

   (c) backward algorithm

   See section 2.3.2 for the full implementation. It follows the implementation initially proposed by Rabiner *et al.* [38].

4. For the purpose of training, a long sequence consisting of transaction symbols from a single cardholder is broken down into smaller sequences of length R. This means we are only looking at the most recent R transactions. The sequences are inputs into the HMM. After learning, an HMM with parameters for a specific cardholder is produced.

5. After the parameters are learned, the model is now ready to classify any new sequence. This is achieved by taking the new sequence, and breaking it down into smaller sequences of length R (in our case 30). Say $O_1, O_2...O_R$ is the path of length R. This in our case will be the first 30 transactions of the cardholder. Then the sequence, which is up to time t, is fed into the HMM. The HMM computes the likelihood α of the path which can be defined as follows;

$$\alpha_1 = (O_1, O_2, O_3...O_R|\lambda) \tag{21}$$

   where λ represents the learned HMM.

6. If O_{t+1}, is the next symbol which represents the next transaction for the cardholder, the next sequence of length R can be formed by appending

observation O_{t+1} to the original sequence O and dropping O_1. The resulting sequence will be $O_2, O_3, O_4, ..., O_{R+1}$. The new sequence can now be fed into the HMM to produce the second likelihood α_2.

$$\alpha_2 = (O_2, O_3, O_4 ... O_{R+1}|\lambda) \tag{22}$$

7. After each α has been calculated, anomalies can be identified by checking whether the likelihood falls within the set threshold. The set threshold can be determined by various strategies with the main ones being by either defining the average likelihood based on the historical transactions or using a cost function set up to minimise loss depending on the risk tolerance of the credit card issuing bank. If the O_{R+1} transaction symbol is accepted to be within the threshold, the transaction is added and persisted to the sequence permanently. Otherwise the transaction is not added as part of the new sequence which is to be trained and used for evaluation. The next incoming transaction can then be evaluated.

The last step above can be repeated for any number of incoming transactions in the sequence or for any number of transaction symbols in the sequence.
In this research, having learnt the HMM parameters as given above with R being set at 30, the learned HMMs were used to evaluate the incoming transactions from simulated data. The results are as shown in Table 1.

3.2.2 HMM Model for All Credit Cardholders

Another approach to modeling credit card transactions will be to train and maintain one HMM for all transactions of various cardholders belonging to one population category. The reason for training from different sequences from the same population is to enable the model to capture an almost complete distribution of the population being trained. Due to incomplete datasets as discussed in 2.4, training a model on one sequence from a single user might mean the learned distribution will not capture the complete distribution of the population category, this will more likely result in the model misclassifying a new sequence from the same population. Everything else remains the same in terms of the training stages of Single HMM per cardholder, as presented in section 3.2.1. However, at stage 4, instead of

training of the HMM from a sequence generated from one cardholder, the HMM is trained on various sequences from different cardholders. The results of training and evaluation from this approach are in Table 1 below.

3.2.3 HMM Model per Category of Credit Cardholders

Alternatively, a single HMM model can also be trained and maintained for various population categories. In this approach detecting anomalies in credit card transactions will be the same as the steps highlighted in one HMM per cardholder, as presented in section 3.2.1 above. However, at step 4 instead of feeding sequences generated from a single cardholder, various sequences from various population categories are fed into the HMM model for training.

The reason for this type of model is to try and explore the impact of different category populations. Various credit card usage patterns exist in credit card user populations. This is due to certain kinds of people sharing the same type of interests and habits. Therefore, it would make sense to train different models for each category and evaluate new instances separately as well. However, one would not know for the incoming transaction sequence which population category it belongs to. This lack of prior knowledge might mean the results from this will vary greatly depending on the HMM which evaluates the incoming transaction. For instance, if a model trained from population A, evaluates an incoming transaction sequence from population A, then the expectation is a higher probability as compared to when the incoming sequence does not originate from the same trained model.

In our case the model training data consists of three categories, category A, B and a mixture of category A and B population user transactions. The results from this training are presented in Table 1, below, for different Cardholders (CH) populations: In Table 1, Single CH, refers to the Single Credit Cardholder model. Population A refers to a model which is trained only from a collection of transactions belonging to one user category in terms of behavioural characteristics. Population A, B, AB is a model in which the training data set is a mixture of various population categories, thereby increasing population diversity per training.

The second approach, HMM Model for All Credit Cardholders, section 3.2.2 demonstrates the same approach with various user data belonging to the

Table 1: Training of Different Cardholder (CH) populations with HMM

Run	Single CH	Category A	Category A, B, AB
'Run-1	86%	84%	60%
'Run-2	90%	92%	56%
'Run-3	64%	96%	56%
'Run-4	86%	84%	58%
'Run-5	92%	94%	44%
'Run-6	72%	76%	58%
Average	82%	88%	56%

same population category. This approach of exposing the model to a lot of transactions from credit card users from the same population category, we believe is responsible for the good performance of the model as on Table 1 in the results column Population Category A. The multi-sequence training data from the user population complements each other with missing data from other user sequences being replaced in other user sequences. Additionally, since HMMs perform better with more data, where there are even more historical transactions belonging to the same user population, the model performs even better as there are more similar instances to learn from. This may increase the probability that the model may learn the true characteristics of the population. This we believe explains the even better performance in HMM Model All Cardholders approach.

However, the same model approach using HMMs tends to degrade in performance when it is applied to credit card transactions belonging to a varied user population as demonstrated in section 3.2.3, HMM Model per Category of Credit Cardholders. This varied user population brings diversity, which in turn introduces complexity and multi-level latent variables that we believe the flat HMMs cannot easily model at least in its standard two-level structure.

3.2.4 Effect of Population Diversity

From the above approaches, the results seem to be suggesting that the performance of HMMs tends to degrade as diversity increases. Initially, as the number of cardholders is increased from 1 cardholder, in the first model to 200 cardholders belonging to same population category as shown in the sec-

ond model in section 3.2.2, the performance of the HMM tends to improve. However, as the categories of the user population increase the performance degrades sharply as shown in the results in Table 1. Diversity in this research refers to different population categories, as classified by the features in the data sets. This diversity brings in data overlap which makes learning more difficult. Data overlap means that behaviour which would otherwise be seen as normal in one population category will be flagged as anomalous in a different population category. This increases the number of false positives. As discussed earlier in section 2, to make a good anomaly detection model, the system needs to have a very low false positive rate.

We believe the reason for the degradation in performance is due to the inability of the flat HMM to model some of the multi-level latent variables with shared hierarchical structures in the diverse populations. These shared features such as user conservatism or high spending habits tend to classify populations more correctly into their respective categories than just bundling them together. The bundling means that some features will not be modelled as effectively as in models which clearly defines hierarchies. In addition, the flat structure means the model is not able to distinguish between the various levels of hidden states which naturally exist in diverse populations. These multi-level variables for credit card user data can include slow transitioning states such as age groups and earnings in higher levels and fast transitioning states such as purchase items category and seasons (weekdays or holidays) in lower levels.

3.3 HHMM based anomaly detection models

In this section, we present anomaly detection in a diverse population based on HHMMs, which we turned to in an attempt to enhance the representation power of Markov chain-based models, especially for multi-category credit card populations. The HHMMs are built from a group of HMMs. We argue that the advantage of a Hierarchical HMM is that it allows for a decomposition of a complex, deep, and diverse population into sub-activities in a course to fine manner which can be learned much more effectively, Wang *et al.* [26, 50]. Our focus will be on credit card transactions that are stochastic with deep multi-latent states and whose observations can be discretised into spending profiles.

We will model stochastic observation from transactions generated by multi-level latent variables in a temporally discrete state space whose structure form various Hidden Markov Models. We also assume that the states are discrete and that the observations are associated with the states which exist in independent forms. Such assumptions have been used before successfully previously where HMMs were used to model anomaly detection in credit card transactions, Singh *et al.* [14, 17, 40].

The following is a basic definition of HHMMs according to the work of Shai *et al.* [16]: a HHMM is defined by the set (S, O, π, A, B, D), where D is the depth of the hierarchy, S, O are the states and the final and /or intermediate observed symbols and π,A, B are the initial, state transitions and observation probabilities.

Depth of the hierarchy describes the number of sublevels before the observation states, so when a normal HMM has $D = 1$, which means the only level of latent states is also responsible for emitting observation symbols. S will be the various hidden states at each level D in credit card modeling. Such hidden states may model the types of goods or services being purchased which will be on lower levels of the hierarchy and other hidden states can comprise of behavioural features of user such as conservative or free spender which will be on higher levels in the hierarchy. Figure 4 illustrates the 3-level hierarchy [**?**]HMM. In contrast to HMMs discussed in section 2.6 and 2.7, HHMMs have the following properties as discussed in [56], which will be important to note for the sake of this research:

1. The hidden state can have different levels. A state S is noted as

$$S_i^d(0 < d <= D, 0 < i <= |S^d|), 1 <= i, j <= N \tag{23}$$

   where d is the level index, i is the state index and S^d is the state at level d. When d=D, s_i^d is called a terminal state

2. Every lower state can have its own standalone HMM with associated initial, transition and emission matrix which activates its child state at s_i^{d+1}

3. The lowest level HMM has observable symbols. The emission probabilities are

$B(s^D) = (b_k(s^D))$, where $b_k(s^D) = P(O_k|s^D)$. For the $d(d < D)$ level HMM, the state sequence in its child could be viewed as its observation. Hierarchical HMM have D levels of HMM which makes it an independent HMM at each level. Each HMM links to its parent and child. In actual fact an HMM model is simply a HHMM model with levels D = 1 according to Fine *et al* [16].

3.4 Enhancements to Tree HHMM

Although a Hierarchical HMM for credit card anomaly detection modeling can have many levels, for the purposes of this thesis we considered one with three levels. The three levels represent most of the possible levels in a HHMM model as far as the frequency of state transitions is concerned. We considered the frequency of transition as the state transition probabilities is the heart of the HHMM and determines how the system behaves. Of these three levels, shown in Figure 3, the top level has slowly changing states; the middle level has occasionally changing states, while the lower level has fast transitioning states. Though more levels could have been considered, three seemed adequate for demonstrating the concept of Hierarchical HMMs without rendering our experiments overly complex.

While neglecting other statistical variables it may be essential to capture other types of statistical structure in user behaviour like spending range, credit user profile (conservative or high spender) [6]. This representation we feel will cater for approximately most dynamics in a diverse population as follows:

1. Top level. Slowly changing states in a credit card scenario represent those variables about an individual which normally change after a long time like job profile; high earner or low earner. Sometimes these might represent more rigid properties which do not change for life for example gender.

2. Middle level represents card holders‘ characteristic which evolve over time but at a much higher pace than top level. This could be such characteristics like on vacation or not, weekends or workdays, public holidays etc.

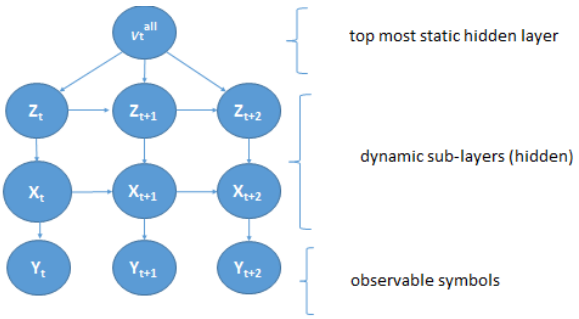


Figure 3: Our version of a Hierarchical HMM

3. The lower level represents the last level before the emission states. The lower level states are the production states responsible for the emitting of observation symbols. These lower states might be changing much more often with each transaction. These states represent those variables such as category of goods being purchased (i.e. groceries or bills, tickets or other smart services).

4. The lowermost level is the observable symbols which are visible from the point of view of the issuing bank. These include the amount purchased, location, merchant type and in some cases other information like the location and time between the purchases. Of course, depending on the device which is being used or the online web application, other variables which are normally hidden can become known and this will improve the model by considering priors to the models.

3.5 Enhanced Model

Two HHMMs were built and trained on synthetically generated sequences representing various transactions from credit card purchases online. The HHMMs are as follows:

3.5.1 Unconstrained HHMM

An unrestricted HHMM can have a variable number of sub states at each upper state. The structure of this HHMM is unconstrained since it allows transitions from any state to any other without restriction. Also, with emitting observables at any level in the hierarchy the model can be referred to as unbalanced. Illustrations of this HHMM are the same as Figure 3, above. The matrix below also shows the transition matrix for such a system.

$$A = \begin{pmatrix} 0.30_{1,1} & 0.30_{1,2} & 0.40_{1,3} \\ 0.40_{2,1} & 0.40_{2,2} & 0.20_{2,3} \\ 0.05_{3,1} & 0.35_{3,2} & 0.60_{3,3} \end{pmatrix}$$

These transitions denote the probability of the state moving from one state to another for a stochastic process which is derived from this model. For example, the first probability, 0,30 denotes the probability of remain in state 0. The moving to the RHS, 0,30 represents the probability of moving from state 0 to 1. Similarly, further on the RHS, the value of 0.4 represents the probability of moving from state 0 to state 2. The important property is that all these probabilities add to 1 since there can be only 3 transitions in this case. The probability of each row should add to 1 as illustrated in Equation 7.

Also, these probabilities were arbitrarily chosen. The initial state transition selected does not affect the accuracy or the performance of the models. This was demonstrated in the experiments to choose the optimal parameters. Each time a different distribution was chosen but each time the performance was the same, so the initial transition probabilities do not affect the model performance.

3.5.2 Constrained HHMM

A restricted HHMM consisting of three levels. However, with an addition that there are certain state transitions between internal states which are restricted which means the transitions probabilities between them will be zero or very negligible. Constrained HHMMs address the modeling of state dynamics by building some topology into the multi-level hidden state representation to make it truly hierarchical [22]. The first level state of this HHMM still had 3 sub states. Each of the states at the third level has 3 child states which produces emissions . Thus, all the emission states are one level. We show the makeup of such a model in Figure 3 above. Also, the

structure seems the same but there is a major difference in the transition matrix with constrained version having zero probabilities between states where transitions are restricted as shown below. A system with three levels and three states at each level can be illustrated by the transition matrix like the one below:

$$A = \begin{pmatrix} 0.00_{1,1} & 0.70_{1,2} & 0.30_{1,3} \\ 0.00_{2,1} & 0.00_{2,2} & 1.00_{2,3} \\ 0.35_{3,1} & 0.00_{3,2} & 0.65_{3,3} \end{pmatrix}$$

The zero probabilities in state $S_{1,1}, S_{2,1}, S_{2,2}$ and $S_{3,2}$ shows that there is a restricted transition from state 1 to 1, 2 to 1, 2 to 2 and 3 to 2.
The generalised Baum Welch algorithm was applied to both HHMMs and the results are explained in section 3.6.

3.6 HHMM learning and Inference

Any HHMM can with multi-levels can be converted in one-level HMM. In such a scenario the probability of moving one state to the always greater than zero which means there is no restriction in state transitions according to Fine *et al.* [16]. The resulting HMM is a flat model without multi-levels. The equivalent HMM lacks, however, the multi-level makeup which we need to use. In our approach we will try and introduce zero probability for transitioning from one state to another if that transition violates the multi-level structure we want to capitalise on.

Illustration of the structure of a HHMM in Figure 4 Aarno [1]. Vertical transitions are in grey lines. Horizontal transitions are in black lines. Internal states are light grey and the dark grey circles are the terminal end states.

The resulting structure of HHMM is very complex as there are both vertical and horizontal transitions which need to be inferred as shown in Figure 4. Depending on the number states at each level and the number of levels in the HHMM, there will be need for us to sum up a lot of combinations for the most likely sequence. This might not be feasible mostly for deep multi-level structured models, as there is an exponential number of combinations which need to be considered.

The alternative will be to convert the HHMM into an equivalent HMM as

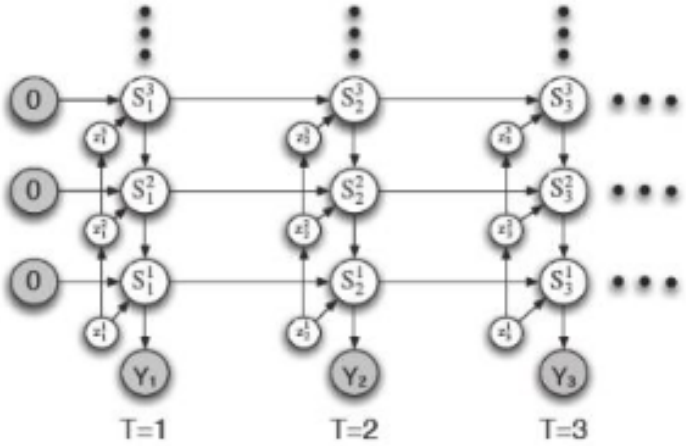


Figure 4: Graphical representation of a Hierarchical HMM, Aarno and Daniel [1]

shown in Figure 6. The resulting structure can then be learned using Baum Welch algorithm. Using dynamic programming it is possible to use a generalised version of the Baum Welch algorithm. The Baum Welch algorithm is an iterative dynamic programming procedure that is used to try find the model parameters (A, B, π) that maximise the probability of an observed sequence $P(V|\lambda)$. The full implementation of this procedure is found in [38]. The Forward–backward algorithm from HMM as explained by [38,49] can be used.

The value of state variables α at level D-1 is equivalent to the value of the HMM which consists only of this level which has observation symbols matrix defined by q_i^D. In order to evaluate the state variables, a recursive bottom up procedure is used in which the sub states value is used to determine the values of the state variable α. Overall, we need to calculate for each state variable the forward algorithm for each observation sequence based on the values of the sub states [16]. This will result in time complexity of order $O(NT^3)$ problem, where N is the total number of states and T is the length of the observation sequence. The $O(NT^3)$ complexity will be better instead of the high time complexity $2O(N^T)T$ in its original direct translation form. In the case of an HHMM the state inference can be done in time complexity represented by a notation of $O(N^3)T$. This inference can be achieved by

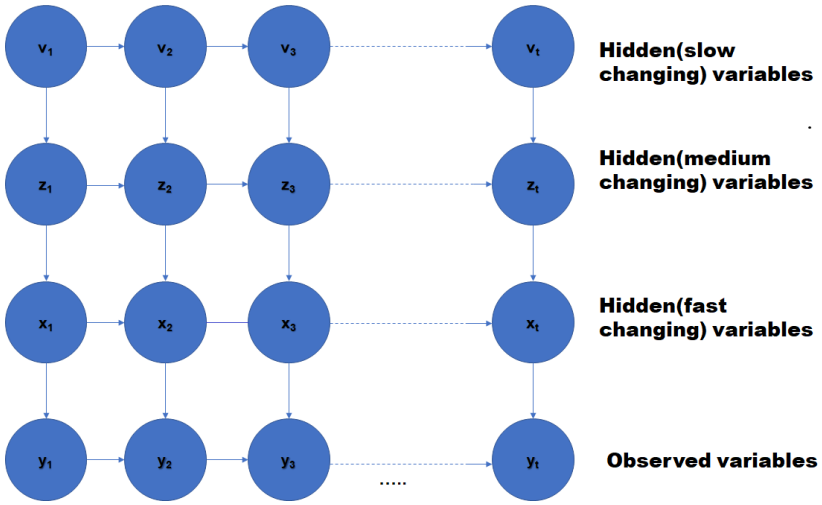


Figure 5: A simple HHMM with three level latent states

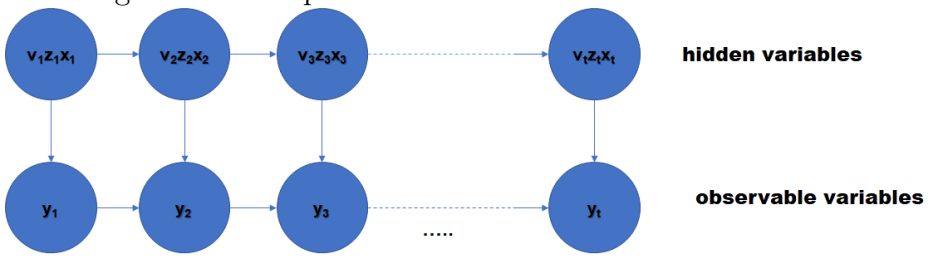


Figure 6: A simple HHMM with three state level converted to an HMM

looping through all possible sequence generated by each sub level HMM [16]. An even better approach will be to treat the HHMM as dynamic Bayesian network (DBN) by unrolling the multilevel states in time [32]. In this DBN representation the latent variables q_i^d at each level, d=1...D, the observation sequence O_t, completely specify the state of the model at time t. This inference produces a complexity of magnitude $O(DTQ^{1.5D}2^{0.5D})$.

In our approach, we employed the generalised forward–backward algorithm for state inference as described in section 2.3.2 and the generalised Expectation Maximization (EM) for parameter estimation based on the forward-backward iterations as outlined by Rabiner *et al.* [38] . This is the same way inference and learning is achieved in HMMs, however, we would need to constraint transitions between levels such that the EM closes the loop in each iteration and preserves the multi-level structure. The EM also learns emission probabilities for the symbols observed with each state, Markov chain probabilities and each sublevel and the inter-level probabilities. The complexity of this algorithm is $O(DTQ^{2D})$, and the full implementation is in [54]. Hence, we use the EM for learning HHMMs, where the model parameters are updated using the EM and this does not affect the convergence especially with a small number of levels [54].

3.7 Conclusion

In this chapter, we looked at approaches to modeling a non-diverse population in the form of single cardholder credit card transactions. We then tried to use the same HMM modeling approach on credit card transactions belonging to diverse credit card user populations. The preliminary results show that the standard HMM does not seem to perform as well as it does in a population of a single category. We believe this seemingly poor performance is due to the fact that an HMM may not be able to capture hierarchies in key structures where model performance might have been enhanced. We then proposed an approach to the enhanced model learning in which naturally occurring hierarchies in the data are exploited. We believe the hierarchies in HHMM captures close to real life hierarchies in credit card transaction data. Real life occurrences come in hierarchies which makes this approach most appropriate. In the next chapter, we will present experiments conducted with the enhanced Hierarchical HMMs, first on single category population and then on diverse populations. We will then build and implement hierarchical

HMMs in modeling anomaly detection in a diverse population.

4 EXPERIMENTS SET UP AND RESULTS

4.1 Introduction

In this section, we will discuss the steps taken in this thesis to set up robust experiments and simulations to test and evaluate the effectiveness of Hierarchical HMMs on diverse populations. As explained in section 3.3, an HHMM is defined by the set (S, O, π, A, B, D), where D is the depth of the hierarchy, S, O are the states and the observed symbols and π, A, B are the initial, state transitions and observation probabilities. Initially, preliminary experiments were run to find the optimal values for the setup of the actual tests. These optimal parameter values include such elements as the length of the training sequence, number of fraudulent instances in a training sequence, observation symbols and sigma which is the threshold variance.

In section 4.2, the artificial data sets are generated by HMMs and HHMMs which are randomly designed to represent Single User Population, Single User Category Population and Diverse Populations using simulated distributions. Section 4.2.4, introduces real data we used to test anomaly detection, which is data generated from a stochastic process with small anomalies in this case the taxi association data from NYC Taxi and Limousine Commission [7]. In the work presented in this thesis, we used the NYC dataset (Taxi dataset) to demonstrate our model on real world data. Thereafter the chapter is structured as follows: Section 4.3 explores the various experimental parameters, selecting optimal values for these parameters. In section 4.4, we explore the standard HMMs and Hierarchical HMMs on various populations generated in section 4.2. We also discuss the results in the same section (4.4) and conclude the chapter in section 4.5

4.2 Data Sets

Testing for anomaly detection can be done in several varying population categories. We have simulated and generated non-diverse and diverse populations data sets synthetically for testing the solution proposed. We developed a simulator that generates normal and anomalous transactions. The anomalous transaction generator has a different distribution function to the normal transaction generator. The transactions are mixed and normally distributed based on the user spending profile. The user spending patterns observations

vary from 0 for the lowest range of transactions to 3 for the highest range of transaction (i.e. low spenders will have more observation symbols of 0,1 while high spenders will have more observation symbols between 2 and 3. For middle spenders, the mean of the observation will be around 1,2).

An example for a specific diverse population with a defined distribution with 12 hidden states and 4 observation symbols. The HHMM generated normal data set is shown in Figure 7. Another example showing potential fraudulent

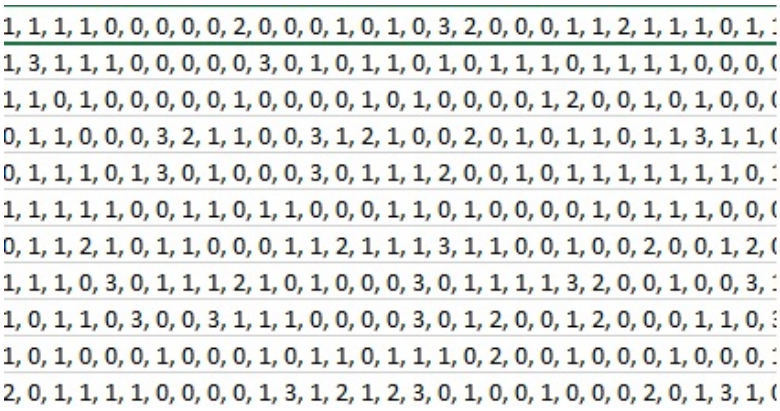


Figure 7: HHMM normal generated data

dataset is shown Figure 8. In Figure 7 a lot of 0 and 1 symbols are found

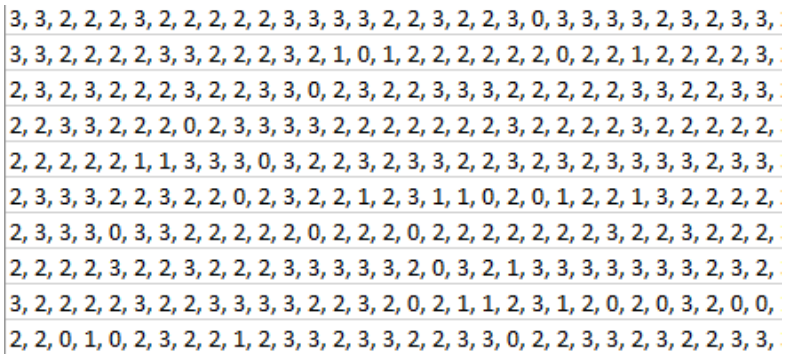


Figure 8: HHMM fradulent data set

in the generated sequences which shows that the particular user/s had most transactions in the lower ranges. The low to medium ranges are represented

by zeros and ones. In Figure 8 a lot of twos and threes in the generated shows the user is a high spender, buying items in the medium to higher ranges. We will give details in the next section on how such data was generated.

4.2.1 Generate HMM Data

The non-diverse data for the experiments was generated to represent real time data from a single user credit card transactions as suggested by various papers in literature such as Srivastava et al *et al.* [13, 14, 44], dhok2012credit and Divya et al. In those papers the authors generated the data from a graphical model λ whose initial parameters were randomly selected, and they are shown in the simulator in Figure 9. The simulator was developed from publicly available Csharp Machine Learning Library Accord Framework.net [43]. The normal data HMM simulator has 2 latent states in which it can produce up to 4 emission symbols. The corresponding fraudulent data simulator also has 2 latent states in which it can produce up to 4 emission symbols. The transition matrix, Emission matrix, and initial state distribution for normal data set A_n, B_n, π_n is as follows:

$$A_n = \begin{pmatrix} 0.9_{1,1} & 0.1_{1,2} \\ 0.1_{2,1} & 0.9_{2,2} \end{pmatrix}$$

$$B_n = \begin{pmatrix} 0.4_{1,1} & 0.4_{1,2} & 0.1_{1,3} & 0.1_{1,4} \\ 0.5_{2,1} & 0.4_{2,2} & 0.05_{2,3} & 0.05_{2,4} \end{pmatrix}$$

$$\pi_n = \begin{pmatrix} 0.3_{1,1} & 0.7_{1,2} \end{pmatrix}$$

And for fradulent data set, the transition matrix A_f, the Emission matrix B_f and the π_f is as follows:

$$A_f = \begin{pmatrix} 0.1_{1,1} & 0.9_{1,2} \\ 0.9_{2,1} & 0.1_{2,2} \end{pmatrix}$$

$$B_f = \begin{pmatrix} 0.03_{1,1} & 0.02_{1,2} & 0.45_{1,3} & 0.5_{1,4} \\ 0.1_{2,1} & 0.1_{2,2} & 0.6_{2,3} & 0.2_{2,4} \end{pmatrix}$$

$$\pi_f = \begin{pmatrix} 0.7_{1,1} & 0.3_{1,2} \end{pmatrix}$$

The values of A_f, B_f were randomly chosen to represent the anomalous behaviour which represented a fraudster sequence of transactions and should generally be different to a normal user. In the case that they behaviour are

the same it will be difficult to pick up the fraud. Emission matrix B_n shows that in both states 1 and 2, there is a high probability of generating 0 and 1 symbol than 3 and 4. For instance the following first ten symbols in sequence 3 show 15 symbols as (2, 3, 0, 1, 0, 1, 1, 1, 1, 2, 0, 0, 0, 1, 0,). Transition matrix for the fraudulent data set which is almost the opposite shows that the model is more likely to produce 2 and 3 as shown by the following example of the fraudulent sequence (3, 2, 2, 3, 2, 2, 3, 2, 2, 3, 2, 2, 2, 0, 2).

The actual steps conducted to generate the data;

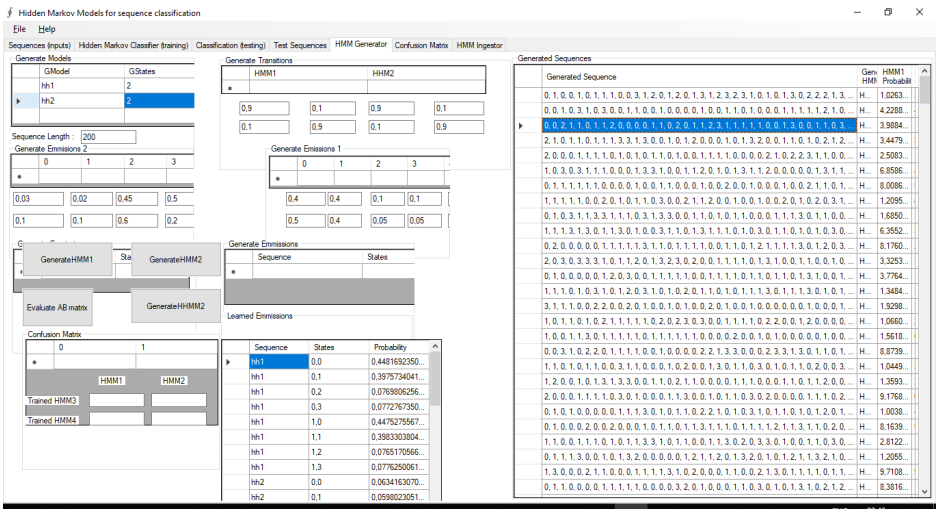


Figure 9: HMM Data Simulator

1. Generate Individual Population A by sampling from a graphical model λ_A in form of an HMM with the probabilistic distribution whose transition matrix is A_n, emission matrix B_n, π_n. Generated 200 sequences of 100 transactions into a flat file to represent a normal card holder population. Generate Individual Population A anomalous data set by sampling from an HMM with probabilistic distribution model (λ_A^f)whose transition matrix is A_f, emission matrix B_f, π_f.

2. Generate Individual Population B by applying a random factor α_R where R is the number of random factors for R datasets being generated. $\alpha_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ to transition and emission probabilities in A to form

λ_B model.

$$A_n^B = \begin{pmatrix} 0.9_{1,1} & 0.1_{1,2} \\ 0.1_{2,1} & 0.9_{2,2} \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

$$A_n^B = \begin{pmatrix} 0.6_{1,1} & 0.4_{1,2} \\ 0.4_{2,1} & 0.6_{2,2} \end{pmatrix}$$

$$A_n^B = \begin{pmatrix} 1.8/2.4_{1,1} & 0.6/2.4_{1,2} \\ 0.2/3.8_{2,1} & 3.6/3.8_{2,2} \end{pmatrix}$$

For emission matrix we also apply a random factor $\beta = \begin{pmatrix} 2 \\ 3 \\ 2 \\ 3 \end{pmatrix}$

to B_n to form B_n^B

$$B_n^B = \begin{pmatrix} 0,4_{1,1} & 0,4_{1,2} & 0,1_{1,3} & 0,1_{1,4} \\ 0,5_{2,1} & 0,4_{2,2} & 0,05_{2,3} & 0,05_{2,4} \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 2 \\ 3 \end{pmatrix}$$

$$B_n^B = \begin{pmatrix} 0.3_{1,1} & 0,2_{1,2} & 0.2_{1,3} & 0.3_{1,4} \\ 0.1_{2,1} & 0.2_{2,2} & 0.1_{2,3} & 0.6_{2,4} \end{pmatrix}$$

To make this a true HMM transition matrix the sum of probabilities in one state should be 1 (section 2.3.2). We apply marginalisation to get

$$B_n^B = \begin{pmatrix} 0.8/2.5_{1,1} & 1.2/2,5_{1.2} & 0.2/2.5_{1,3} & 0.3/2,5_{1.4} \\ 1.0/2.25_{2,1} & 1.2/2,25_{2.2} & 0.10/2.25_{2,3} & 0.15/2.25_{2,4} \end{pmatrix}$$

$\pi_n = \begin{pmatrix} 0.3_1 & 0.7_2 \end{pmatrix}$ α and β values were arbitrarily chosen and applied to the transition and matrix in order to have a new distribution pattern which is different from the previous one. The new distribution represents a different spending profile possibly belonging to another population category. Repeat the generation of 200 sequences with a length of 100 symbols to represent a different category of a user population (e.g. User population B). Apply the same random factors on A_f and B_f to make A_f^B and B_f^B to form model λ_B^1 to generate the anomalous transactions for population B.

3. Generate Individual Population C by applying a random factor α to

emission probabilities in step (2) to form λ_C model. The new distribution represents a different spending profile. Repeat the generation of 200 sequences with a length of 100 to represent a different category of a user. Apply the random factor β to form model λ_C^1 to generate the anomalous transactions for population C.

4. Repeat the steps 2 and 3 until there are 16 different individual populations that represent 16 different categories of cardholder users. Although there can be any number of population categories in a diverse population, for the purposes of this thesis we considered one with 16. Using 16 sequences seemed adequate for demonstrating the concept of diverse populations without rendering our experiments overly complex. The 16 data sets form the diverse population since they are coming from 16 distinctive distributions.

4.2.2 Data Randomness

The initial HMM matrices A, B, and π for both the normal and fraudulent user transaction sequences were chosen randomly and to show that our proposed model performance does not depend on any specific dataset, we also tested various populations whose matrices were generated randomly by using a random number generator. The resulting matrices was then used to generate datasets which were used to train the 3 models (HMM, Unconstrained HHMM and Constrained HHMM). The data used for training was generated as follows:

1. Generate random numbers using a random number generator for each value in the matrices.

2. Marginalise the matric vectors to make it a true HMM transition matrix as discussed in section 4.2.3 item 2. This is achieved by adding all the numbers and then dividing by the total as with A_n^B. For example, if the random generated numbers were 4, 4, 1, 1 and 5, 4, 5, 5

   $$B_r = \begin{pmatrix} 4_{1,1} & 4_{1,2} & 1_{1,3} & 1_{1,4} \\ 5_{2,1} & 4_{2,2} & 5_{2,3} & 5_{2,4} \end{pmatrix}$$

   The resulting matrix will be

   $$B_r = \begin{pmatrix} 4/10_{1,1} & 4/10_{1,2} & 1/10_{1,3} & 1/10_{1,4} \\ 4/18_{2,1} & 4/18_{2,2} & 5/18_{2,3} & 5/18_{2,4} \end{pmatrix}$$

3. Step 1 and 2 is also done for the other probability matrices where the initial values are randomly generated and marginalised to create a true HMM matrix whose probabilities add to 1 for each state.

4. Use the resulting matrices to initialise an HMM which generate training data for a normal user.

5. Repeat step 1 up to 4 for generating the fraudulent sequence data sets.

6. Repeat the steps to generate 11 different data sequences which represents randomly generated data sets from 11 different user population groups.

7.

The data generated from the above steps was used to test whether the proposed models can give the same results from a any dataset or specifically only if its generated from the matrices specified in section 4.2.3. The results of such an experiment are shown in the Table 2 below. Also, the average precision and accuracy per model was calculated. As shown in table 2 of results,

Table 2: Base HMM vs HHMM vs handHHMM models performance on randomly generated data

	HMM	HMM	HHMM	HHMM	handHHMM	handHHMM
Datasets	Precision	Accuracy	Precision	Accuracy	Precision	Accuracy
1	0,870	0,94	0,870	0,94	0,869	0,94
2	0,607	0,72	0,607	0,72	0,607	0,72
3	0,850	0,88	0,607	0,88	0,894	0,9
4	0,222	0,4	0,222	0,4	0,423	0,52
5	0,947	0,94	0,947	0,94	0,809	0,86
6	1,000	0,98	1,000	0,98	1,000	0,98
7	0,889	0,74	0,889	0,74	1,000	0,74
8	1,000	0,98	1,000	0,98	1,000	0,98
9	0,200	0,92	0,750	0,84	0,750	0,84
10	1,000	0,84	1,000	0,84	1,000	0,84
11	1,000	0,84	1,000	0,98	1,000	0,98
Average	0,78	0,83	0,81	0,84	0,85	0,85

there are instances where the model perform poorly on data from certain

datasets for instance dataset 4 where there accuracy percentage is 40, 40, 52 respectively for the 3 models; HMM (base HMM), HHMM proposed by Stavasta *et al.* [45], and Unconstrained HHMM proposed by Fine *et al* [16] and our proposed Constrained HHMM (handHHMM) . However, average performance is consistently showing better performance of the Constrained HHMM. This shows that the better performance of the proposed model is not related to a specific matrix with certain values or specific dataset.

In a real-life example of anomaly detection in streaming, online applications, for example, there will be many normal transactions with a few instances of abnormal ones. The normal transactions would have been generated in line with a stochastic process in normal conditions. In a credit card scenario, the normal transactions will have been generated from a normal user profile. For the purpose of learning user behaviour amounts will be divided into 4 classes of low, lower medium, upper medium and high as discussed in 2.3.2. In our case, we generated the data belonging to these categories which represent more or less credit card transaction data as proposed by Srivastava *et al.* [45].

4.2.3 Generate HHMM Diverse Data

The second data set for the experiments was generated from an HHMM graphical model λ whose initial parameters were learned from a randomly selected online credit card sample. In Figure 10 we demonstrated how HHMM generated data sets have multi-modes and why it will be the most suitable to use to generate diverse population data. This is due to HHMMs having a multi-level latent variable whose state transitions can produce observation which is very varied and with many modes. The following steps were conducted to generate diverse data sets;

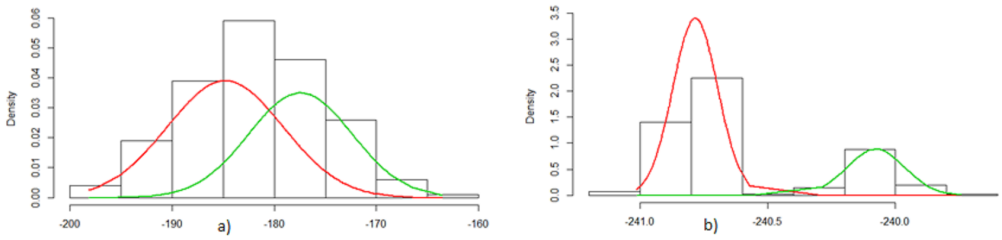
1. Generate Individual Population A by sampling from a graphical model λ_A in form of an HHMM with a given probabilistic distribution that is specific to a particular spending profile as learned from an online sample. Generated 200 sequences of 100 sequences long into a flat file to represent a normal cardholder transaction. Calculate the inverse of the emission probabilities of λ to come up with a second HHMM model λ_A^1 and use it to generate the perceived anomalous data transactions for population A.

2. Generate data as per Step (2) in section 4.2.1 except this time the generating graphical model is HHMM

3. Generate data as per Step (3) in section 4.2.1 except this time the generating graphical model is HHMM

4. Repeat the steps above until there are 16 different individual populations which represent 16 different categories of cardholder users. The 16 data sets form a diverse population.

   From the generated sequences for HHMM and HMM, probabilities and log-likelihood of the sequences were calculated. The log-likelihood vs the density function was plotted in Figure 10. The resulting plot shows the data plotted in a normal distribution to show modes, we can see that one category of HHMM generated data is multimodal with two modes on -240.7671 and -240.0907. The data generated from HMM has only one mode -184.90. This shows that HHMM data is more diverse than HMM. The multimode shows that although the data sets are from one distribution, the parameters responsible for generating the data can have two distinctive behavioural patterns which belong to two populations. The histogram shows the log-likelihood of the sequences generated by HMM and HHMM models. The red line shows the first mode and the green line shows the second mode.



(a) Example of a distribution from an HMM model generated data. The histogram shows that the distribution is normal with one mode

(b) Example of the distribution from a HHMM model generated population shows a possible multimode distribution which is diverse consisting of several modes

Figure 10: HMM non diverse and HHMM diverse distributions

4.2.4 Real Data

For the purpose of checking how our model performs with real-world data. We found datasets on Kaggle which were generated from some stochastic processes. We chose the NYC Taxi and Limousine Commission passenger data. The NYC taxi dataset from Kaggle consists of the aggregated total number of NYC taxi passengers for a period from 01/07/2014 to 31/01/2015 which represents 214 days. The data has five anomalies for the NYC marathon, Thanksgiving, Christmas, New Year's Day, and a snowstorm. The anomalies represent instances during the 214 days where there were very high total number of passengers recorded or extremely low numbers due to a holiday or an event such as marathon day or storm. The data file included here consists of aggregating the total number of taxi passengers into 30-minute buckets. For the purpose of feeding the data into the HMMs model, we first categorized it into 8 buckets as shown in Table 3, NYC Taxi Association Passenger data.

Table 3: NYC Taxi Association Passengers data per 30 minutes interval categorised

Total number of passengers	Number of Observations	Category
8-5007	1536	0
5008-10007	998	1
10008-15007	1416	2
15008-20007	3884	3
20008-25007	1937	4
25008-30007	544	5
30008-35007	3	6
35008-40007	2	7

We used this dataset to validate our models, however for the purposes of choosing HMM factors for our experiments we used the synthetic data. This is because the simulated data allowed us to test various factors without restrictions. In section 4.3, we elaborate how we choose the factors and consonants for the models using simulated data from 4.2.1 and 4.2.3.

4.3 Choosing HMM Factors and Consonants

Getting real bank credit data to test fraud cases was very difficult. In general, banks normally do not agree to share their data with academic researchers. Additionally, we could not find any benchmark credit card fraud data for our experiments hence we performed various simulation runs to generate our own data. We, therefore, performed experiments with the HMM generated data in Figure 4a), which is a single sequence non-diverse population. We use this data set to set up the following HMM factors; number of anomalous transactions (4.3.3), learning sequence length (4.3.4), number of observation symbols (4.3.1), Sigma (i.e. the value which is to be used as the threshold in setting the range of normal and anomalous transaction 4.3.5) and number of states (4.3.2).

4.3.1 Observation symbols

In real life, ranges of transaction amounts can be used as symbols and the types of items can be considered as states. Any HMM can, theoretically, have any number of symbols. The model could use any clustering algorithm such as K-means to dynamically allocate each price to a range. Most papers used 3 observations for the value of M (number of symbols), Phua*et al.* [14,35,52]. Our baseline HMM arbitrarily used 3 observation symbols [46], in our case since we were generating the discretised symbols synthetically, we choose 4 observation symbols. The value of four was carefully selected in order to have enough symbols to represent the various price ranges. We have therefore used the observation symbols 0,1,2,3 to represent the transactions ranges as per different spend categories as follows:

1. Symbol 0, represented the lowest amounts purchased using the credit card and this is the lower quartile range;

2. Symbol 1 represented the lower medium range of amounts that could be purchased using a credit card;

3. Symbol 2 represented the upper medium range of amounts that could be purchased using a credit card; and

4. Finally, symbol 3 represented the final quartile of transaction amounts. These are high, unusual amounts which are seldom done by cardholders unless for big items like TV, cars, etc.

We believe our selection will adequately cater for all pricing ranges that can be there in a population particularly for the purpose of demonstrating our model.

4.3.2 Number of hidden states

In any HMM choosing the number of states usually is quite critical and might determine the performance of a model. In real life, the number of states can be chosen depending on the prior knowledge of the system being modeled. For instance, in our case, we anticipated that we have 3 layers of hidden variables, the fast transitioning, medium transitioning and slow transitioning levels as described in section 3.4. In our case, we tried to minimise the levels to three and three states at each level for the sake of demonstration but as already alluded to in section 3.3, there can be no restriction to the number of states that can be modeled. For the baseline HMM we used 10 hidden states as per the experimental results [46]. Additionally, the number of states was increased to 12 in the more diverse HHMM experiments in order to capture all the dynamics of varied population.

4.3.3 Number of Anomalies

In order to determine the optimal number of anomalies with which the model can start detection effectively, we ran several simulations with all other factors constant and varying the number of abnormal instances. In order to simulate a transaction sequence that is anomalous, we took a normal sample sequence of length 100 from population A normal data set generated in section 4.2.1. We also took a sample sequence from the population A fraudulent data set generated in section 4.2.1. The normal sequence is split into two sequences of 50 lengths each. One of the split sequences is used to train an HMM model, and the other one is used for testing. In that second sequence used for testing, a few consecutive anomalous transactions are drawn randomly and injected at position 30 of the second sequence of length 50. The second sequence of length 50 forms the anomalous sequence and is used for testing the trained sequence.

The injection was done at position 30 to allow the model to settle by first modeling 29 good transactions before it starts modeling anomalous ones. This simulated dataset with the anomalous instance is highly similar to the

NYC taxi dataset as discussed in 4.2.4, which has 5 anomalies. In the first experiment run, only 2 anomalous transactions are inserted into the second sequence. An HMM model is then trained on the normal first sequence. Testing is done on the second anomalous sequence. This first experiment with 2 anomalous transactions is repeated 20 times and the average accuracy of the 20 runs is recorded. Ultimately, the average result is then recorded in a table of accuracy vs Fraud Sequence length. In the second experimental run, 3 anomalous consecutive transactions are used for testing. In the third and fourth experiment runs, 4 and 5 anomalous transactions are used for testing the trained model and the average results per 20 runs are also recorded.

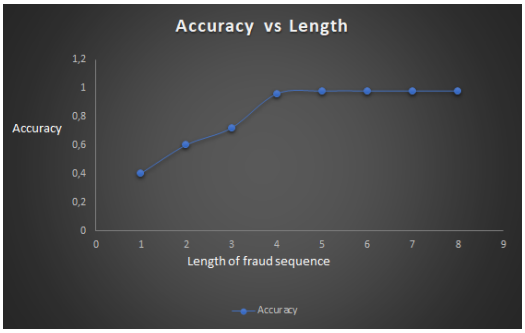


Figure 11: Optimal Length of Anomalous transactions occurrence in HMM model

Figure 11 shows the plot of Fraud Sequence Length vs Accuracy. The plot shows that the model only starts to perform better when there are 4 or 5 consecutive anomalous transactions. After these experiments, we then concluded to use 5 anomalous transactions in testing the performance of HMMs and HHMMs on diverse populations in all other experiments that were conducted.

From this experiment and the results, it can be concluded that a small number of anomalous transactions make it difficult for the model to detect anomalies. This is due to the imbalanced data sets in anomaly transaction detection. Imbalance exists when there are fewer anomalous transactions from which the model can learn enough so as to be able to pick the anomaly. This explains why the model performs better with 4 and/or more than with

1 and 2 abnormal transactions in the testing sequence. Our submission is that:

1. For less than 4 consecutive anomalous transactions, the performance is poor. However, for the time-series data to work for the models discussed in this thesis there has to be a minimum of 4 consecutive fraudulent data points. This can be a situation whereby a fraudster is able to make those first one or two transactions before the model alerts the authorities of the anomalies developing. To bring this into perspective, the anomaly data set on Kaggle which we used to test this model on real data, has time series data collected in 30 mins intervals for 214 days. The interesting part is that although the anomalies are given as 5 different days (Day 125, 150, 178, 185 and 211), there are two important observations:

   (a) An anomaly of a day is actually 48 data points of anomalies (30-minute intervals forms 48 data points in a day). This is more than 4 data points we require for our model to predict.

   (b) The fraudulent points were always in a continuous sequence which also work well for the model.

2. Alternating or one normal transaction in the middle worked fine in our experiments as long as there were at least 3 other transactions that were abnormal as shown in 4. The model was able to perform okay in those circumstances. The models struggled on performance where there were only one or 2 anomalous transactions. Five or more anomalous transactions produced better performance. Unfortunately, without other attributes about an anomalous transaction our model was not able to perform effectively to pick the anomaly.

In table 4 below, run 5 and 6 has 4 and 2 normal transactions in between anomalous sequence respectively. This causes the performance to decrease. However, run 8 and 9 has zeros (which is normal) inserted alternatively in the sequence and this seem not to affect the model performance. This shows that even with alternating or one normal transaction in the middle of anomalous sequence the model still can pick the anomalies effectively. However, having more than one normal transaction in between the anomalies will cause the model performance to reduce in comparison to when you have 1.

Table 4: Model Performance with normal data point in between anomalies and or at the end

Run	Normal Sequence	Anomalous Sequence	Accuracy	Recall
1	0, 0, 0, 2, 0, 0, 0, 1, 1, 2, 0, 0	2,3,2,2,2,1	96	95
2	0, 1, 0, 0, 2, 1, 0, 0, 1, 2, 1, 0	3,2,2,2,2,2	90	60
3	1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0	2,3,3,2,3,3	97	90
4	0, 1, 0, 2, 0, 1, 2, 2, 1, 1, 1, 1	3,2,3,3,2,2	87	90
5	0, 1, 0, 3, 1, 1, 1, 3, 3, 2, 2, 2	3,3,1,2,0,1	30	0
6	1, 1, 3, 1, 0, 0, 1, 0, 1, 0, 1, 1	2,2,0,3,3,2	71	60
7	0, 0, 0, 0, 1, 2, 1, 1, 1, 0, 1, 0	1,2,3,3,1,3	90	85
8	2, 2, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0	3,2,0,3,3,1	81	93
9	1, 1, 0, 2, 0, 0, 1, 1, 1, 2, 1, 1	0,3,0,1,2,2	97	95
10	0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2	3,3,3,3,0,2	79	93

4.3.4 Sequence length

For the purpose of determining the optimal sequence to be used for HMM learning, two experiments were conducted. The other factors were kept constant, 4 symbols, 4 instances of anomalous transactions. In the first experiment, we wanted to explore KL divergence value vs the sequence length. In the second experiment, we explore the Probability range vs the sequence length. The data that was generated (4.2.1), was used for both experiments. The data consisted of 200 sequences that have 100 transactions each.

Exploring the KL divergence : The KL-divergence, [33] also known as the relative entropy, between two probability density functions f(x) and g(x), is commonly used in statistics as a measure of similarity between two density distributions. The KL divergence is used in many aspects of speech and image recognition, such as determining if two acoustic models are similar. We had 50 sequences(4.2.1) of normal transactions with 100 transactions each. If one sequence is to be used for both training and testing, for example, to select (an) optimal sequence length as shown in our case, it would have to be broken into two sequences as in (4.3.3). Therefore one of the 100 transactions were split into two sets with 50 transactions each, one for testing (anomalous) and one for training (normal). To enable effective training the 50 transactions are split further into even smaller sequences of equal lengths which are finally fed into the HMM. For example, if we choose the length of 20 for the smaller

sequence, we built up the data set to be fed into the HMM as follows:

1. Take the transactions from positions 1 up to 20 from the normal sequence, and add it into an array
2. Take the second set of data with transactions from positions 2 up to 21
3. Take the third set of data with transactions from position 3 up to 22
4. Repeat steps (2) and (3) until the last set of data from position 31 to 50
5. Combine all the arrays from all the steps to form a two-dimensional array to be ingested into the model for training. An example of such an array that was used for training is shown in Figure 12.

Generated Sequence	Generated HMM	HMM1 Probability
2, 0, 0, 0, 0, 1, 1, 1, 0, 2, 1, 3, 0, 0, 1, 3, 1, 1, 3, 0	HHM1	6,75000538610306E-12
0, 0, 0, 0, 1, 1, 1, 0, 2, 1, 3, 0, 0, 1, 3, 1, 1, 3, 0, 1	HHM1	3,32942214114261E-11
0, 0, 0, 1, 1, 1, 0, 2, 1, 3, 0, 0, 1, 3, 1, 1, 3, 0, 1, 2	HHM1	5,74860962912706E-12
0, 0, 1, 1, 1, 0, 2, 1, 3, 0, 0, 1, 3, 1, 1, 3, 0, 1, 2, 1	HHM1	5,0367656375251E-12
0, 1, 1, 1, 0, 2, 1, 3, 0, 0, 1, 3, 1, 1, 3, 0, 1, 2, 1, 1	HHM1	4,42750666852519E-12
1, 1, 1, 0, 2, 1, 3, 0, 0, 1, 3, 1, 1, 3, 0, 1, 2, 1, 1, 1	HHM1	3,88259943867004E-12
1, 1, 0, 2, 1, 3, 0, 0, 1, 3, 1, 1, 3, 0, 1, 2, 1, 1, 1, 0	HHM1	4,41168041421706E-12
1, 0, 2, 1, 3, 0, 0, 1, 3, 1, 1, 3, 0, 1, 2, 1, 1, 1, 0, 0	HHM1	5,02576503857949E-12

Figure 12: A normal sequence with 50 transactions after being converted to a 2 dimensional array with each array of length 20 which is fed into an HMM model for training

Figure 12 shows part of the resulting two-dimensional array after breaking the original sequence of length 50. The figure only shows 8 sequences used to build up the complete array for training. In the figure the first column is the actual sequence, the second column is the training HMM and the last column is the calculated probability after training using the model in column 2. More arrays of length 20 could be added until all transactions in the original 50 transactions long array have been added. After training each of the sequences will have their own probabilities and log-likelihoods. These are used as the values for the first probability density function. The second

probability density function is generated using the same procedure with the only difference being that the data would have been generated by the second HMM. In our case, we used data from the fraudulent HMM for category A population(4.2.1).

In order to select the optimal length of the array to be used this experiment was done with data sets from category A, with a length of 10,15,20,30,40 and 50. For each experiment 20 runs were conducted at each length and the results were averaged and recorded in a table. The results were plotted against KL divergence value of the model. The results show that good performance and distinction between two HMMs is obtained by considering the sequence of length 20 or more. This is shown in Figure 13 below.

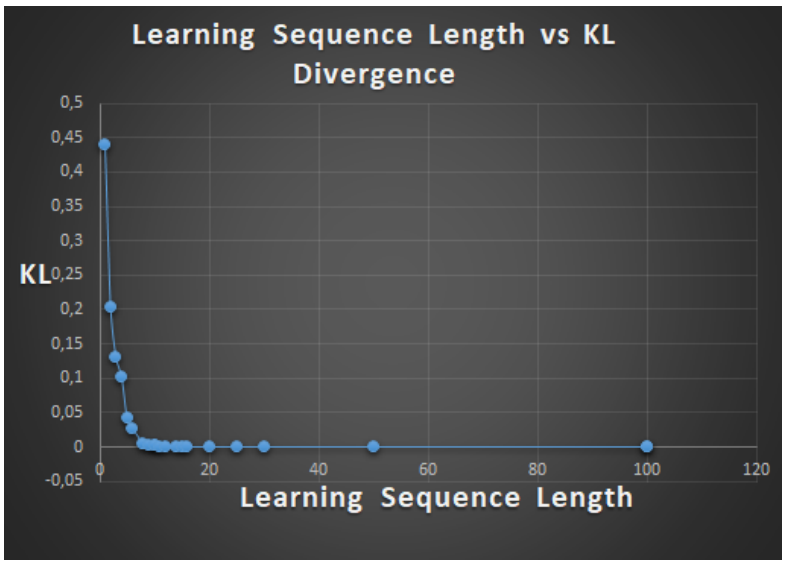


Figure 13: Optimal Length of Learning Sequence

The KL divergence plot has been used to determine at what sequence length will we get a good comparison between two HMMs. Figure 13 shows that any sequence with the length of 20 or more will be a reasonable length to use.

Explore Likelihood Range : After training of any of the sequences by an HMM the learned model can be used to calculate the probability of the

likelihood of occurrence of a given sequence of observation. This probability P(O—λ), where O is the sequence and λ is the model, was discussed in section (2.3.2). Since the calculated probabilities for the sequences are very low (the first sequence in Figure 12 is 6,75000538610306E-12, its log-likelihood would be -11,17), a log-likelihood of the probability will be better to use particularly if we need to compare the probability to another parameter (sequence length) in a graph. Log-likelihood is generated by finding the log of the calculated probabilities from each sequence. It was with this background that the second experiment was conducted to identify the optimal sequence length at which the model will be more stable and give a more consistent probability range for sequence array built from the same sequence. In other words, we looked at the results which gave us the least difference between the upper and the lower bounds of the average probability of the trajectory.

Figure 14 below shows the graph of the log-likelihood range vs Sequence length. The reason why we took the lowest value in the range is because the lower the range the more sensitive the model is and the thinner the threshold. A thinner threshold (smaller log-likelihood range) means we are less likely to have a lot of false positives than a wide range which might allow some anomalous trajectories to be misclassified as normal. Table 5 shows the data used to plot the Likelihood Range(Loglikelihood of the array sequence probability) vs Sequence Length (training model sequence) graph. The table also shows the actual Range (i.e. the difference between the highest and the lowest log-likelihood for a given set of 50 sequences). It also shows the Depth of the array sequence (the number of sequences that were used in the model during evaluation). From both Table 5 and Figure 14, it can be seen that the

Table 5: Likelihood range vs Sequence length

Sequence length	Sequence Depth	Likelihood Range	Range
15	50	-08 to -12	7
20	50	-09 to -12	7
30	50	-14 to -17	3
35	50	-12 to -18	6
40	50	-46 to -51	7
50	50	-49 to -56	8

optimal threshold range of magnitude 3 occurs at sequence length 30. There-

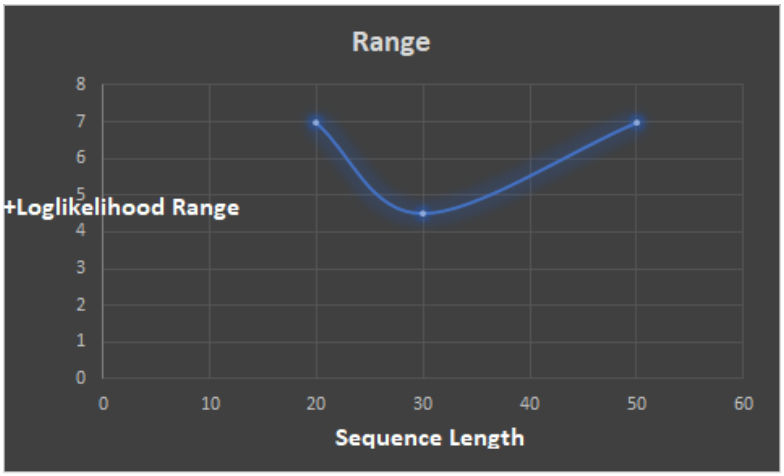


Figure 14: Loglikelihood range Learning Sequence

fore, in all our experiments the sequence length was set at 30 transactions for optimal performance.

4.3.5 Choosing sigma

To detect anomalies banks normally set a threshold which is the minimum probability that a detection system allows before it flags a transaction(s) as suspicious. In this thesis, we defined the threshold as the average probability of non-fraudulent sequences. The averaging minimises errors of measurement and smoothen the results. As shown in figure 12 this probability fluctuates with a given range. Sigma in this experiment is defined as the variance between the average and the lower bound of the threshold.

We had 50 sets of sequences as in the previous experiment 4.3.4. Because HMMs should be fed with smaller array sequences we also had to break down the 50-transaction sequence into 50 sequences as we did in 4.3.4. We fed the model with varying length of array sequence at different sigma values (0.5, 1.0, 1.5, 2.0, 2.5, 3.0) as shown in Table 6. We then plotted the values of sigma vs accuracy. We ended up with a graph shown in figure 15. This showed us that there is a possibility for the maximum performance of the model at Sigma of 1.

Our design factors were as follows:

Table 6: Sigma Model Test Selection Results

Model	Accuracy	Recall (T P/ (T P +F P))	‘Sigma
HMM	88%	100%	3
HMM	94%	100%	2
HMM	98%	95%	1
HMM	98%	95%	.5

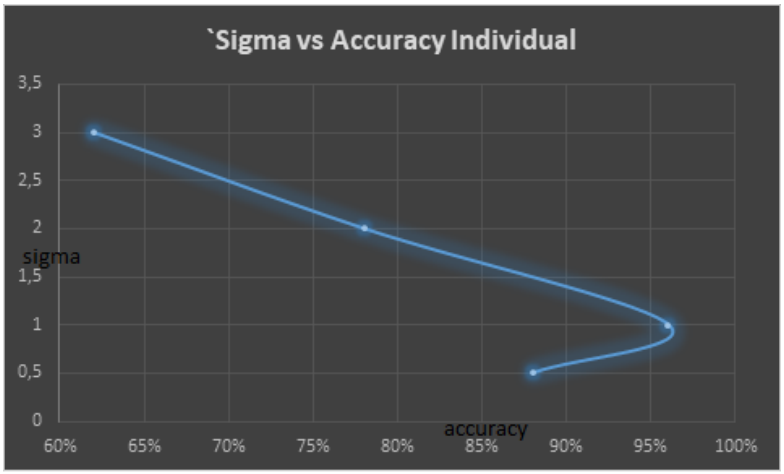


Figure 15: Sigma Selection

1. Number of latent states N, 10 for baseline HMM and 12 for the HHMM models;
2. Sequence length 30;
3. Threshold value for the log-likelihood 1; and
4. Number of observation symbols 4.

4.4 Experiments and Results

Before exploring the HHMM performance in diverse populations, it is necessary to understand the definitions of the various populations we have generated as explained in section 3.2 :

1. Population Category A consists of HMM simulated data sequences belonging to the same type of population called category A.

2. Population Category B consists of multiple HMM simulated data sequences belonging to the same type of population called category B.

3. Individual Population A consists of a single HMM simulated data sequence belonging to a single user from Category A.

4. Individual Population B consists of a single HMM simulated data sequence belonging to a single user from Category B.

5. HMM Diverse Population consists of sixteen distinct population categories generated from 16 different HMM distributions.

6. HHMM Diverse Population consists of sixteen distinct population categories generated from 16 different HHMM distributions.

These populations were used in various scenarios in the next section (4.4.1).

4.4.1 Scenario 1: Single Sequence HMM

We first test a single sequence which is generated as elaborated in section 4.2.1, single category population A. In this case, a single sequence with 200 observations are broken down into two sets. The first 50 observations from the training set and observation 84 to 143 forms part of the validation set. Both the training and the validation set sequences are 30 observations long as discussed in section 4.3.4. This could represent a scenario where the HMM model is trained from data coming from one individual of population category type A. After running the 20 experiments, the individual HMM tends to perform better than the population based HMM (generated datasets representing various sequences from different cardholders but belonging to the same category of uniform population, 3.2.2). This could be because in an individual-based population there is less diversity and the model is more likely to reject any new behaviours which do not fit with what it has already seen. The downside of this is that when there is overlapping the model is more likely to miss that anomalous transaction that looks normal. Overall, when compared to population-based, the performance was better as shown in Table 7.

Table 7: Single vs Multiple Sequence HMM

Population	Accuracy Recall (T P/(T P +F P))	Standard Dev
Individual Population A	98%	98%	5%
Full-Population Category A	98%	98%	5%
Individual Population B	86%	76%	13%
Full-Population Category B	98%	98%	9%

4.4.2 Scenario 1: Multi Sequence HMM

Secondly, we train the HMM model using multiple sequences generated from a single category of the user population, for instance, A population. The only difference with the experiment in the above section 4.4.1 is that in here the HMM is being trained by observations from multiple sequences. Intuitively this could be a case of taking transaction data of users with the same behavioural characteristics. This is then followed by training of the HMM model on this data and use the learned model to predict abnormal behaviour in newly observed data.

From Table 7 we can see that training from a single sequence of observations from user population category A gives the same accuracy for both HMM and HHMM. However, the training on population B shows that there is massive improvement in accuracy from 86 to 98. The increase in the accuracy on category B population can be explained by the fact that there is more data to train on from category B user population form which the model can better than during the individual category B user sequence training and therefore there was a chance to improve. The training from full dataset from population B category gives the model a chance to learn comprehensively the full distribution of the model and therefore there is increased performance.

4.4.3 Scenario 2: Single Sequence Unconstrained HHMM

In scenario 2, we test the performance of an Unconstrained HHMM (3.5.1), which are structured multi-level stochastic processes. HHMMs generalise HMMs by making each of the hidden states a self-contained probabilistic model which is itself an HMM. The Unconstrained nature means the state transitions can occur between any state to another without restriction. This is different from constrained HHMM which has restriction between certain states, the transition matrix can then be constrained to allow transitions

only between specific states, a property explained in 3.5.2, which preserves the hierarchical topology of the model. In an Unconstrained HHMM, the model is trained using a single sequence from a population generated by a diverse HHMM generator as elaborated in section 4.2.3. Intuitively, one can think of this experiment as testing data from a population of various diverse categories such as HHMM Diverse Population(4.4). However, only one population category type is selected for training, making this experiment the same as Scenario 1 Single HMM in section 4.4.1. The only difference is that in this case, the training model is an Unconstrained HHMM instead of a standard HMM. The HHMM is fed with observations from the individual category population. After training, the trained HHMM is then used to predict anomalies in an incoming transaction sequence from an unknown user from the same population category as the training. The results are shown in Table 8.

4.4.4 Scenario 2: Multi Sequence Unconstrained HHMM

In scenario 2-part b, we test the performance of an Unconstrained HHMM which is trained from multiple sequences from the HHMM Diverse Population. The Unconstrained nature means the state transitions can occur between any other state to another without restriction. The model is trained using multiple sequences from a population generated by a diverse HHMM generator as elaborated in section 4.2.3. Intuitively, one can think of this experiment as testing data from a population of various diverse categories. The HHMM is fed with a diverse population consisting of 128 sequences (8 sequences per each category), each sequence with 100 transactions. The transactions are represented by symbols between 0 and 3 inclusive as we discussed in section (4.3.1). The model has three latent levels and three states at each level.

After training, the learning HHMM is then used to predict anomalies in an incoming transaction sequence from an unknown user. The unconstrained HHMM tend to show superior performance in both time and accuracy to the HMM counterpart as shown in Table 8 below. The results of an unrestricted HHMM model were almost similar to the individual HMM but showed improvement in the population based HHMM trained models. This could be because HHMMs captures more characteristics which it can correlate to when the complexity and diversity of the model increases. However, the time to

run this model was also considering this could be because since the model is not restricted it runs through all the permutations, in terms of executing all the internal state transitions at each level. The time complexity increases by a factor of D every time a level is added. Therefore, the time complexity of evaluating the values for all states of an HHMM is $O(NT^3)$, where N is the total number of states and T is the length of the observation sequence. [16].

4.4.5 Scenario 3: Multiple Sequence Constrained HHMM

In scenario 3, we test the performance of a restricted HHMM while using multiple sequence data from a diverse user population. The restrictive nature is fully explained in section 3.5.2 which means the state transitions cannot occur between any state to another arbitrary state but rather with restriction as defined by the hidden states transition probability matrix. The model is trained using multiple sequences from a population generated by a diverse HHMM generator as elaborated in section 4.2.3. A simple intuition of this might be for one to think of this experiment as testing data from a population of various diverse categories. The HHMM is fed with 128 sequences (8 from each of the 16 diverse populations generated in section 4.2.3). Each sequence has 100 transactions represented by symbols (0,1,2,3). The transition matrix (A_{con}) with zero probabilities between certain states representing the restrictions being enforced on that transition path for example ($0.00_{1,3}$) means there is no transition from state 1 to state 3 since the probability for that happening is zero. After training, the trained HHMM is then used to predict anomalies in an incoming transaction sequence from an unknown user from one of the 16 category populations.

Table 8 shows the accuracy percentages of the three models (HMM(53),

Table 8: Accuracy for HMM, Unconstrained HMMM vs Constrained HHMM

Population	HMM	Unconstrained	Constrained	Standard Dev
Individual Population A	81%	82%	82%	15%
Population Category B	78%	80%	82%	15%
HMM Diverse Population	76%	78%	78%	13%
HHMM Diverse Population	53%	58%	63%	12%

Unconstrained HHMM(58) and Constrained HHMM(63)). The models were tested on 4 populations; Individual Population A, Population Category B,

HMM Diverse Population, and HHMM Diverse Population. We can see that learning diverse populations is relatively difficult as shown by the generally low percentage accuracies for all three models HMM (53), Unconstrained HHMM(58) and Constrained HHMM(63)). However, as the diversity decreases (in HMM Diverse Population) the relative performance of the models improves and is even much better on a single category population. In Table 8 it can also be seen that the actual results improve per model from a standard HMM to Unconstrained HHMM and finally the Constrained HHMM has the best results. Table 8 shows the experimental results. The results in table 8 show that although the complex diverse population has lower probabilities, relatively the Constrained HHMM performs better than the HMM and Unconstrained HHMM. So, this seems to suggest that as diversity increase the learning becomes more complex. The Table 16 and Figure 16 combine all the diverse experiments to show the performance of base HMM, HHMM and Unconstrained HHMM. The results of the experiments are shown in Figure 16. The tabular results are also shown in the table 9.

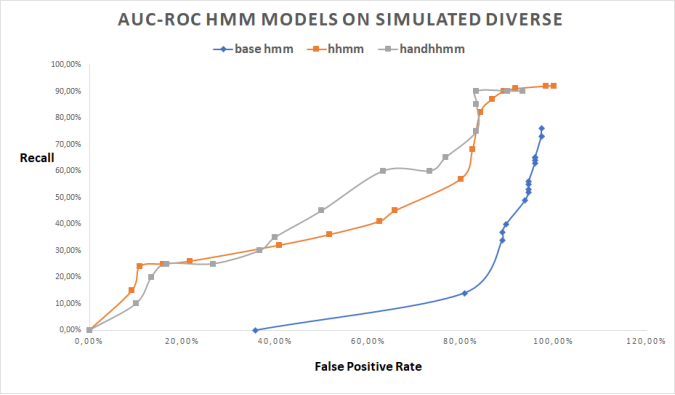


Figure 16: Simulated Diverse Data Results AUC - ROC Results

Figure 16 and Table 9 shows the Recall vs FPR as the threshold is being varied for HMM models. The results are evaluated using AUC (area under the curve) of the ROC (Receiver Operating Characteristics) as discussed by Provost et al [38]. The ROC gives a more reliable performance comparison in anomaly detection than just looking at Accuracy. This is also further explained in [47]. From the ROC graphs it shows that our proposed model Constrained HHMM (referred to as handHHMM in the graph) outperforms the standard HMM (base HMM in the graph) as proposed by Srivastava

et al. [46] and the Unconstrained HHMM (HHMM in the graph) which is a state-of-the-art Hierarchical model as proposed by Fine *et al.* [16]. The Table 9 shows the main results of experiments discussed in this thesis. The table

Table 9: Base HMM vs HHMM vs handHHMM models performance on Simulated Diverse Data

	base HMM	base HMM	HHMM	HHMM	handHHMM	handHHMM
Threshhold	Recall	FPR	Recall	FPR	Recall	FPR
1	76	97	100	92	93	90
2	73	97	100	92	93	90
3	73	97	98	92	90	90
4	65	96	92	91	90	90
5	65	96	89	90	83	90
6	65	96	87	87	83	85
7	64	96	84	82	83	75
8	63	96	83	68	77	65
9	56	95	80	57	73	60
10	55	95	66	45	63	60

shows the main results of work presented in this thesis.

4.4.6 Scenario 4: NYC Taxi Data test with base HMM, HHMM and handHHMM

In this scenario we tested the performance of the basic HMM model on anomaly detection as per the Srivastava paper [46], we then test in against the HHMM and our improved HHMM model which we call handHHMM. Before feeding the data into the model we performed some pre-processing as discussed in 4.2.4. We categorised the data into 8 distinctive sets forming a discretised set with 8 observations from 0 to 7 which means the full set of symbols are Symbols=0,1,2,3,4,5,6,7. Since the data is actually the total aggregated number of passengers in every 30-minute intervals for New York City, each value corresponds to a category/symbol from between 0 and 7 as in table 3. This forms a time series data with 48 data points per day for the 215 days. The 48 daily data points can easily form a sequence of length 48. Since the anomalies are identified in days it was easy for us to limit the sequence length to 48 as well such that the model will the fed with 215 data points.

We also divided the 216 points into training and test sets, in our case, we trained using 70 percent and 30 percent as the testing set. The sequences are fed into the base, HMM, HHMM and handHHMM and the testing is done against the 30 percent test set. The results are also evaluated using AUC of the ROC. The results of the experiments are shown in Figure 17.

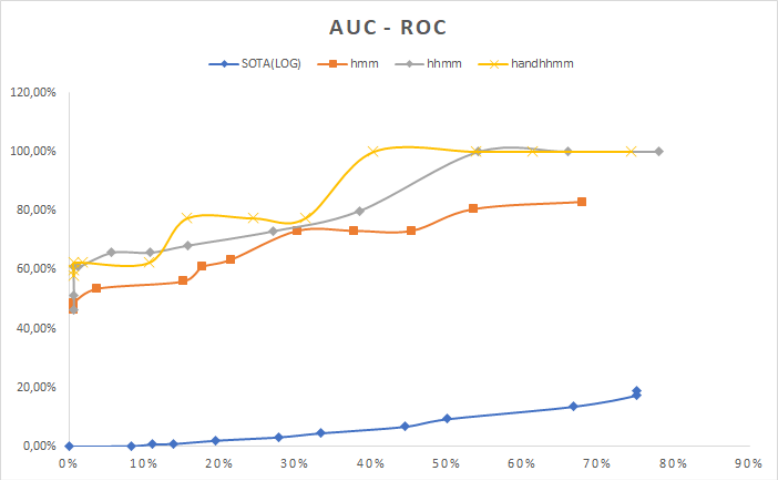


Figure 17: NYC Taxi Data AUC - ROC Results (real data)

The tabular results are also shown in the table 10. Figure 17 and Table 10 shows the performance of Recall and FPR as the threshold log-likelihood of the sequences is varied from -0.5 to 16,6. The insights from the graph are as follows: although the recall starts at 0.63 percent for all models the FPR starts at 46 percent for base hmm and standard HHMM respectively, the FPR for handHHMM starts at a higher value of 58 percent at a broader threshold of 16.5 and the FPR steadily rises to 60, 62 77 and 100 percent showing superior performance at all times as compared to the baseline HMM model and standard HHMM. There are however instances where handHHMM is outperformed at threshold 3. Seems the optimum performance for the handHHMM occurs at threshold 1 with Recall at 40 percent and FPR at 100 percent. Overall, the graphs show that handHHMM performs better than the baseline HMM and HHMM. From the ROC graphs it shows that our proposed model handHHMM outperforms the base HMM Srivastava *et al.* [46] and the standard HHMM [16].

This shows our improved HHMM will be able to correctly predict any given

Table 10: Base HMM vs HHMM vs handHHMM models performance on NYC Taxi Data

	LOG	LOG	baseHMM	baseHMM	HHMM	HHMM	handHHMM	handHHMM
Sigma	Recall	FPR	Recall	FPR	Recall	FPR	Recall	FPR
-0,50	75	19	68	83	78	100	74	100
0	75	18	53	80	66	100	61	100
0,50	67	14	45	73	54	100	54	100
1,00	50	9	38	73	38	80	40	100
1,50	44	7	30	73	27	73	31	78
2,00	33	5	21	63	16	68	24	78
2,50	28	3	18	61	11	66	16	78
3,00	19	2	15	56	6	66	11	63
4,00	14	1	4	54	1	61	2	63
4,50	11	1	1	49	1	61	1	63
13,50	8	0	1	46	1	51	1	60
16,50	0	0	1	46	1	46	1	58

data point randomly selected from the normal or anomalous classes better than the others. We have also included a non-graphical method for anomaly testing techniques in the form of logistic regression on the real data to show how it performs against our method. As shown in Figure 17, the LOG does not perform well as compared to the graphical models. This we believe shows that for the set of data with the characteristics and structure of time series with latent variables, it is better modeled by a graphical model in this case in form of HMMs. Additionally, our enhanced constrained HHMM (hand-HHMM) shows better performance.

4.5 Conclusion

This section described the experiments conducted in this thesis to try and compare a normal HMM to hierarchical HMM. Firstly, sequences for generating sequential data are illustrated in section 4.2. Then the setting up of the experimental parameters follow in sections 4.3.1, 4.3.2, 4.3.3, 4.3.4, 4.3.5. After the setting of the model, the normal HMM is trained and tested with diverse and non-diverse populations data sets. Similarly, the Constrained and Unconstrained HHMMs are also trained and tested with diverse and non-diverse populations data sets.

In the experiments, we have learned two main populations, diverse and non-diverse, with two main models, HMMs and hierarchical HMMs. The results show the effectiveness of the HHMMs in both diverse and non-diverse populations. This effectiveness observed in Hierarchical HMMs can be explained by the fact that hierarchical models have multi-level latent variables that are able to pick up the various distributions and complexities in diverse populations. They then express those model complexities and distributions in terms of graphical manipulations, in which the underlying mathematical expressions are carried along implicitly and comprehensively. The results further show that Constrained HHMMs are more effective in learning diverse populations than Unconstrained HHMMs. This could be explained by the fact that a simple constraint on the transition parameter of an HHMM can successfully capture the slowly transitioning latent variable and fast transitioning multi-latent variables in complex diverse data and the true topology of the hidden state representation. The captured true topologies and multi-level latent variables can then effectively model the varied patterns and complexities present in diverse population data sets. In the following chapter, we conclude this thesis.

5 CONCLUSION AND FUTURE WORK

In this thesis, we have proposed the use of HHMMs for anomaly detection in diverse populations. We have discussed an approach to anomaly detection by constraining transition matrix parameters to limit certain state transitions. We have also shown an approach to our experimental design parameter setting. We compare the performance of our approach with the baseline hmm credit card anomaly detection method proposed by Abhinav *et al.* [44]. We have shown how the HHMMs performance is mostly superior as we vary the diversity of the simulated populations. ROC analysis also has shown that our proposed method performs better as compared to the baseline method.

5.1 Domain Application

In the financial domain, especially in auditing, computer-assisted audit tools and techniques (CAATs), allows auditors to extract and analyse transactions from enterprise resource planning (ERP) systems for anomalies. The CAATs which now use some data mining techniques such as HMM models discussed in this thesis generates in most cases a lot of anomalies that needs to be followed up and investigated. There is usually a cost associated with analysing and following up each anomaly identified. Analysing false positives is time-consuming and consequently causes costs without the realisation of benefits. It is, therefore, imperative that the anomaly detection tool/technique being implemented should have low false positives.

Furthermore, in anomaly detection, particularly credit bank card fraud, the cost of missing a true positive is equally catastrophic as this may result in the customer losing money, leading to financial losses to the institution. It is therefore equally important that any successful fraud detection technique being selected by the bank will be able to detect the costliest fraudulent transaction. The risk tolerance of the bank or financial institution should determine the threshold value under which the fraud system should raise a fraud alarm.
Through our analysis and experiments performed in this thesis and the interpretation of the results with the use of the area under the curve for the ROC, we believe we have laid a good base from which other researchers or financial institutions which want to implement an anomaly detection techniques can leverage on. They can use the analysis in this paper as a basis to choose an

anomaly detecting technique and determine the threshold values which will be in line with the company risk tolerance for fraud.

5.2 Conclusion

Anomaly detection in diverse populations with complex structures remains an untapped territory. An adequate solution will be a valuable addition to the broader field of anomaly detection such as financial card fraud and for exploring anomaly detection in diverse populations. We provided a probabilistic, hierarchical approach to anomaly detection in diverse populations which extends standard HMM, by attempting to restrict certain transitions between hidden states.

We verified the single sequence population testing using both the HMM and the HHMM. HMM was shown to perform better on a single population. When the Unconstrained HHMM was used it did not perform any better than a standard HMM. Therefore, for a single population, the best performance comes from a standard HMM. For a multi-sequence population, the experiments validated that an Unconstrained HHMM outperforms the standard HMM. Also, in a population made up of a mixture of population categories, the Unconstrained HHMM performs better than the standard HMM. It was also validated that in all diverse population the Constrained HHMM outperforms both standard HMM and Unconstrained HHMM.

Lastly, we showed that diverse and complex data from varied populations is the most difficult to learn, however, the good news is that the results show constrained HHMM tends to learn the complex data set faster and with better performance than both baseline HMM and Unconstrained (standard) HHMM. Based on the observations as summarised in Table 9, it can be concluded that:

1. The richer graphical models with multi-levels and hierarchy tend to outperform the simple flat models.

2. The diversity of the population introduces data overlaps which result in increased misclassification. To a certain extent, the introduction of richer models improves the chances of preserving the performance of the models.

3. The diverse populations complicated the learning of unconstrained models resulting in performance degradation.

We also conclude on the type of data sets that the HMM models discussed in this thesis will work effectively, as follows:

1. Number of consecutive anomalous transactions is 4 or more.
2. Data is sequential time series data which can be easily discretised.
3. The structure of the transactions is stochastic in nature.
4. The latent states are hierarchical.

5.3 Future Work

We have demonstrated the models on real data from the NYC Taxi Association. However, due to the limitation of access to financial data from banks or financial institutions, the HHMM approaches proposed in this thesis were not tested on financial institutions data, future work can focus on that area if the data becomes available. Additionally, another feature that will be worth adding is the application of the anomaly testing to other domains which are non-financial domains such voice recognition and protein DNA decoding for further analysis of its application and performance with the aim of identifying anomalies that might indicate certain symptoms. This could be a very valuable contribution to the literature of anomaly detection, particularly in diverse populations. Further studies of the approach put forward in this thesis would potentially form a valuable contribution to the literature on anomaly detection, particularly in diverse populations.

References

[1] Daniel Aarno. *Intention recognition in human machine collaborative systems*. PhD thesis, KTH, 2007.

[2] Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal. Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68:90–113, 2016.

[3] SABRIC SOUTH AFRICA. Card fraud 2015.

[4] Phillip L Ainsleigh, Nasser Kehtarnavaz, and Roy L Streit. Hidden gauss-markov models for signal classification. *Signal Processing, IEEE Transactions on*, 50(6):1355–1367, 2002.

[5] V Bhusari and S Patil. Study of hidden markov model in credit card fraudulent detection. *International Journal of Computer Applications (0975–8887) Volume*, 2011.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[7] Patrick L Brockett, Richard A Derrig, Linda L Golden, Arnold Levine, and Mark Alpert. Fraud classification using principal component analysis of ridits. *Journal of Risk and Insurance*, 69(3):341–371, 2002.

[8] ed. Bryan Garner. Black's law dictionary. 8th ed. (2004), s.v., "fraud.", 2004.

[9] Hung H Bui, Dinh Q Phung, and Svetha Venkatesh. Hierarchical hidden markov models with general state hierarchy. In *Proceedings of the national conference on artificial intelligence*, pages 324–329. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[10] Philip K Chan and Salvatore J Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. 1998.

[11] Khyati Chaudhary, Jyoti Yadav, and Bhawna Mallick. A review of fraud detection techniques: Credit card. *International Journal of Computer Applications*, 45(1):39–44, 2012.

[12] Martin Cooke, Phil Green, and Malcolm Crawford. Handling missing data in speech recognition. In *Third International Conference on Spoken Language Processing*, 1994.

[13] Shailesh S Dhok and GR Bamnote. Credit card fraud detection using hidden markov model. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(1):231–237, 2012.

[14] Sneha Janardhan Dhanashree Rathod Amruta Sardeshmukh Divya.Iyer, Arti Mohanpurkar. Credit card fraud detection using hidden markov model. *International Journal of Computer Applications (0975 – 8887)*, 45:333–338, 2012.

[15] Tom Fawcett and Foster Provost. Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3):291–316, 1997.

[16] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

[17] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):9–42, 2001.

[18] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International journal of pattern recognition and artificial intelligence*, 15(01):9–42, 2001.

[19] Zoubin Ghahramani and Michael I Jordan. Factorial hidden markov models. In *Advances in Neural Information Processing Systems*, pages 472–478, 1996.

[20] Sushmito Ghosh and Douglas L Reilly. Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, volume 3, pages 621–630. IEEE, 1994.

[21] Jaakko Hollmen. Probabilistic approaches to fraud detection. *Licentiate's Thesis, Helsinki University of Technology, Department of Computer Science and Engineering*, 15, 1999.

[22] Michael I Jordan and Chris Bishop. An introduction to graphical models, 2004.

[23] M Kavitha and M Suriakala. Fraud detection in current scenario, sophistications and directions: A comprehensive survey. *International Journal of Computer Applications*, 111(5):35–40, 2015.

[24] Min-Jung Kim and Taek-Soo Kim. A neural classifier with fraud density map for effective credit card fraud detection. In *Intelligent Data Engineering and Automated Learning—IDEAL 2002*, pages 378–383. Springer, 2002.

[25] Angelika I Kokkinaki. On atypical database transactions: identification of probable frauds using machine learning for user profiling. In *Knowledge and Data Engineering Exchange Workshop, 1997. Proceedings*, pages 107–113. IEEE, 1997.

[26] Weicong Kong, Zhao Yang Dong, and David J Hill. A hierarchical hidden markov model framework for home appliance modelling. *IEEE Transactions on Smart Grid*, 2016.

[27] Yufeng Kou, Chang-Tien Lu, Sirirat Sirwongwattana, and Yo-Ping Huang. Survey of fraud detection techniques. In *Networking, sensing and control, 2004 IEEE international conference on*, volume 2, pages 749–754. IEEE, 2004.

[28] Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. Credit card fraud detection using bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, pages 261–270, 2002.

[29] SABRIC Kanyisa Ndyondya Communications Manager. 2017 card fraud booklet, protect your card and information at all times, 2017.

[30] Uzi Murad and Gadi Pinkas. Unsupervised profiling for identifying superimposed fraud. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 251–261. Springer, 1999.

[31] Kevin Murphy. A brief introduction to graphical models and bayesian networks, 1998. *Available electronically at http://www. cs. ubc. ca/ murphyk/Bayes/bnintro. html*, 2004.

[32] Kevin P Murphy and Mark A Paskin. Linear-time inference in hierarchical hmms. *Advances in neural information processing systems*, 2:833–840, 2002.

[33] Vittorio Perduca and Grégory Nuel. Exact computation of kullback-leibler distance for hidden markov trees and models. *arXiv preprint arXiv:1112.3257*, 2011.

[34] Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *Acm sigkdd explorations newsletter*, 6(1):50–59, 2004.

[35] Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010.

[36] Louis Strydom Partners Johannesburg. PwC Global Malcolm Campbell, Gerhard Geldenhuys. Global economic crime and fraud survey 2018, the dawn of proactivity: Countering threats from inside and out, 2018.

[37] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[38] Lawrence R Rabiner and Biing-Hwang Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

[39] S Benson Edwin Raj and A Annie Portia. Analysis on credit card fraud detection methods. In *Computer, Communication and Electrical Technology (ICCCET), 2011 International Conference on*, pages 152–156. IEEE, 2011.

[40] Anshul Singh, Devesh Narayan, et al. A survey on hidden markov model for credit card fraud detection. *International Journal of Engineering and Advanced Technology (IJEAT)*, 1(3), 2012.

[41] Marios Skounakis, Mark Craven, and Soumya Ray. Hierarchical hidden markov models for information extraction. In *IJCAI*, pages 427–433, 2003.

[42] Padhraic Smyth. Belief networks, hidden markov models, and markov random fields: a unifying view. *Pattern recognition letters*, 18(11-13):1261–1268, 1997.

[43] Cesar Souza. Hiddenmarkovclassifier class — Waccord, the machine learning library, 2016. [Online; accessed 22-July-2016].

[44] Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun Majumdar. Credit card fraud detection using hidden markov model. *IEEE Transactions on dependable and secure computing*, 5(1):37–48, 2008.

[45] Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun K Majumdar. Credit card fraud detection using hidden markov model. *Dependable and Secure Computing, IEEE Transactions on*, 5(1):37–48, 2008.

[46] Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun K Majumdar. Credit card fraud detection using hidden markov model. *Dependable and Secure Computing, IEEE Transactions on*, 5(1):37–48, 2008.

[47] S Stolfo and AL Prodromidis. Agent-based distributed learning applied to fraud detection. Technical report, Technical Report CUCS-014-99, Columbia Univ, 1999.

[48] George Tauchen. Finite state markov-chain approximations to univariate and vector autoregressions. *Economics letters*, 20(2):177–181, 1986.

[49] Anton Tenyakov. *Estimation of Hidden Markov Models and Their Applications in Finance*. PhD thesis, University of Western Ontario, 2014.

[50] Limin Wang, Yu Qiao, and Xiaoou Tang. Latent hierarchical model of temporal structure for complex activity classification. *IEEE Transactions on Image Processing*, 23(2):810–822, 2014.

[51] Panhong Wang, Liang Shi, Beizhan Wang, Yuanqin Wu, and Yangbin Liu. Survey on hmm based anomaly intrusion detection using system calls. In *Computer Science and Education (ICCSE), 2010 5th International Conference on*, pages 102–105. IEEE, 2010.

[52] Shiguo Wang. A comprehensive survey of data mining-based accounting-fraud detection research. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, volume 1, pages 50–53. IEEE, 2010.

[53] Wei Wang, Xiao-Hong Guan, and Xiang-Liang Zhang. Modeling program behaviors by hidden markov models for intrusion detection. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, volume 5, pages 2830–2835. IEEE, 2004.

[54] Lexing Xie, Shih-Fu Chang, Ajay Divakaran, and Huifang Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden markov models. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 3, pages III–29. IEEE, 2003.

[55] Masoumeh Zareapoor, KR Seeja, and M Afshar Alam. Analysis on credit card fraud detection techniques: Based on certain design criteria. *International Journal of Computer Applications*, 52(3), 2012.

[56] Hua-Ping Zhang, Qun Liu, Xue-Qi Cheng, Hao Zhang, and Hong-Kui Yu. Chinese lexical analysis using hierarchical hidden markov model. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 63–70. Association for Computational Linguistics, 2003.

[57] Yang Zhou. Structure learning of probabilistic graphical models: a comprehensive survey. *arXiv preprint arXiv:1111.6925*, 2011.

[58] Zahra Zojaji, Reza Ebrahimi Atani, Amir Hassan Monadjemi, et al. A survey of credit card fraud detection techniques: Data and technique oriented perspective. *arXiv preprint arXiv:1611.06439*, 2016.