CREDIT RISK ANALYSIS USING HIDDEN MARKOV MODEL

by

Hasan Tahsin Oğuz

B.S., Electrical and Electronics Engineering, Bilkent University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Systems & Control Engineering
Boğaziçi University

2009

CREDIT RISK ANALYSIS USING HIDDEN MARKOV MODEL

APPROVED BY:

Prof. Fikret Gürgen (Thesis Supervisor)	……..
Prof. Levent Akın	……..
Asst.Prof. Murat Saraçlar	……..

DATE OF APPROVAL: 21. 01. 2009

ACKNOWLEDGEMENTS

I would like to emphasize my special thanks to thesis supervisor, Prof. Fikret Gürgen who offered this nice thesis topic to me. Many insightful suggestions and helpful comments during this study with him are also appreciated.

I would also express my special thanks to Prof. H. Levent Akın and Asst. Prof. Murat Saraçlar for their valuable participation to my thesis jury.

I am also grateful to Mrs Oya Aran for her vital suggestions in my thesis and encouragements about HMM.

Finally, I would like to express my deepest thanks and love to my family and my dear angel Dr. Melek Boynukalın.

ABSTRACT

CREDIT RISK ANALYSIS USING HIDDEN MARKOV MODEL

The purpose of this study is to investigate the performance of Hidden Markov Model (HMM) for credit risk analysis in terms of classification and probability of default (PD) modeling where PD modeling assigns default bankruptcy probabilities to credit customers instead of strictly classifying them as good (solvent) borrowers and bad (insolvent) borrowers. PD modeling process makes use of some data belonging to credit applicants and helps banks or credit companies compute a probability that the customer should not pay his/her debts in a timely manner, prior to the decision of granting credit. In the first part of this study, classification ability of HMM is compared to that of Logistic Regression (LR) and k-Nearest Neighbors (k-NN) which are two conventional methods for classification. In the second part, PD modeling performance of HMM is analyzed and compared to that of LR which is known to be one of the most popular PD modeling methods so far. Australian Credit Database and German Credit Database are two public datasets utilized in this thesis study. Classification performances of the aforementioned methods are judged according to accuracy, error cost and Receiver Operating Characteristics (ROC) analysis with supporting experiments using six-fold cross validation. PD modeling performances of HMM and LR are also compared by directly examining the average PD values for solvent and insolvent borrowers. Matlab's HMM toolbox by Kevin Murphy is used for HMM computations whereas a web based tool is utilized for LR analysis. Matlab is also used for k-NN analysis in classification experiments. The aim of this study is to build appropriate algorithms for HMM to make it an effective way of credit risk analysis as well as conventional methods. The results of the experiments show that HMM is a powerful and effective method for credit risk analysis and can be utilized by financial institutions.

ÖZET

SAKLI MARKOV MODELİ İLE KREDİ RİSK ANALİZİ

Bu çalışmanın temel amacı kredi müşterilerini kesin bir şekilde iyi (kredi borcunu geri ödeyen) ya da kötü (kredi borcunu aksatan) diye ayıran bir sınıflandırma yöntemini ve bu yöntemden farklı olarak müşterilerin kredi borcunu geri ödememe (iflas etme) olasılığını hesaplayan temerrüt olasılığı (TO) yöntemini kullanarak Saklı Markov Modeliyle kredi risk analizi yapılabilirliğinin araştırılmasıdır. TO süreci, kredi müşterilerinin bazı verilerini kullanarak kredi verme ya da vermeme kararından önce müşterinin kredi borcunu zamanında ödememe olasılığını hesaplamak suretiyle kredi verilmesi kararı konusunda banka ve kredi şirketlerine ışık tutar. Bu çalışmanın ilk bölümünde HMM yönteminin sınıflandırma kabiliyeti geleneksel yöntemler olan Lojistik Regresyon (LR) ve Yakın k Komşu (YkK) yönteminin sınıflandırma kabiliyetiyle karşılaştırılmaktadır. İkinci bölümde ise HMM yöntemiyle TO modellemesi yapılmış ve oluşturulan modelin performansı en popüler yöntemlerden biri olan LR kullanılarak oluşturulmuş TO modellemesiyle karşılaştırılmıştır. Serbest erişime açık olan Alman ve Avusturalya Kredi Veri tabanları bu tez çalışmasında kullanılmıştır. Yukarıda bahsedilen yöntemlerin sınıflandırma işlemindeki etkinlikleri doğruluk, hata maliyeti ve Alıcı İşletim Karakteristiği (AİK) ölçütlerine göre altılı çapraz doğrulama testleriyle belirlenmiştir. LR ve HMM yöntemleriyle oluşturulan TO modellerinin performansları ise iyi ve kötü müşterilerin ortalama TO değerlerine bakılarak belirlenmiştir. Matlab'ın Kevin Murphy tarafından geliştirilen HMM paketiyle HMM analizleri yapılırken, LR analizleri için internet tabanlı LR hesaplayıcı kullanılmıştır. YkK analizi için yine Matlab kullanılmıştır. Bu çalışmada HMM yöntemi kullanarak geleneksel yöntemlerle rekabet edecek düzeyde kredi risk analizi yapmaya yarayan algoritmaların bulunması hedeflenmiştir. Deneylerin gösterdiği kadarıyla HMM oldukça etkin bir kredi risk analizi yöntemidir ve finansal kuruluşlarca da kullanılabilir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF TABLES ix
1. INTRODUCTION 1
1.1. Credit Risk 1
1.2. Objectives 4
1.3. Organization of the Thesis 4
2. CREDIT CLASSIFICATION 6
2.1. Credit Data 6
2.2. Classification with Hidden Markov Model 9
2.2.1. Hidden Markov Model 10
2.2.2. Proposed Method for HMM Classifier 14
2.2.3. Implementation of Proposed Method 16
2.3. Classification with Logistic Regression 25
2.3.1. Logistic Regression Model 25
2.3.2. Classification of Credit Customers with Logistic Regression 27
2.4. Classification with k Nearest Neighbors 30
2.4.1. k - Nearest Neighbors Model 30
2.4.2. Classification of Credit Customers with k-NN 32
2.5. Classification Performances and Experiments 34
2.5.1. Classification Accuracy 35
2.5.2. Classification Error Cost 38
2.5.3. Receiver Operating Characteristics (ROC) 41
2.6. Triage Strategy 42
2.6.1. Triage Strategy in HMM 43
2.6.2. Triage Strategy in Logistic Regression 46
3. PROBABILITY OF DEFAULT MODELING PROCEDURE 49

3.1. PD Modeling with Logistic Regression 50
3.2. PD Modeling with HMM 52
3.3. Alternative Way of PD Modeling by HMM 55
3.4. PD Modeling Performances 56
4. CONCLUSION 59
APPENDIX A: DOCUMENT FOR HMM TOOLBOX OF MATLAB 62
REFERENCES 64

LIST OF FIGURES

Figure 1.1. Credit Risk Analysis Procedure 4

Figure 2.1. Markov Chain with three states and corresponding transitions 11

Figure 2.2. Observations influenced by hidden states in a HMM structure 12

Figure 2.3. Illustration of observation and state sequence for proposed model 15

Figure 2.4. HMM Classification Algorithm 17

Figure 2.5. Logistic Curve 26

Figure 2.6. Classification process of Logistic Regression 28

Figure 2.7 k-Nearest Neighbors Algorithm 31

Figure 2.8. ROC Space 41

Figure 3.1. The concept of PD modeling 50

LIST OF TABLES

Table 2.1. The Character of German Dataset 8

Table 2.2. The Character of Australian Dataset..... 9

Table 2.3 HMM Classification scheme for first subset of German dataset..... 19

Table 2.4. Results of HMM experiments on German Dataset..... 20

Table 2.5. Results of HMM experiments on Australian Dataset..... 20

Table 2.6. Results of HMM experiments on German Dataset for different ordering..... 21

Table 2.7. Results of HMM experiments on Australian Dataset for different ordering.... 22

Table 2.8. Transition Matrix of 'good model' for third subset of German Dataset..... 23

Table 2.9. Transition Matrix of 'bad model' for third subset of German Dataset..... 24

Table 2.10. Results of Logistic Regression experiments on German Dataset..... 29

Table 2.11. Results of Logistic Regression experiments on Australian Dataset..... 29

Table 2.12. Results of k-Nearest Neighbors experiments on German Dataset..... 33

Table 2.13. Results of k-Nearest Neighbors experiments on Australian Dataset..... 33

Table 2.14. Contingency Table for Binary Classification 36

Table 2.15. Classification Accuracy of HMM, LR and k-NN for German Dataset..... 37

Table 2.16. Classification Accuracy of HMM, LR and k-NN for Australian Dataset 37

Table 2.17. Cost Matrix of Two Class Classification for Credit Applications..... 39

Table 2.18 Error Costs 40

Table 2.19. ROC Distances..... 42

Table 2.20. Results of Triage Strategy of HMM on German Dataset 45

Table 2.21. Results of Triage Strategy of HMM on Australian Dataset 45

Table 2.22. Results of Triage Strategy of Logistic Regression on German Dataset 47

Table 2.23. Results of Triage Strategy of Logistic Regression on Australian Dataset..... 48

Table 3.1. PD Modeling Results with HMM..... 56

Table 3.2. PD Modeling Results with Logistic Regression..... 57

Table 3.3. Summary of PD modeling results for both datasets 58

1. INTRODUCTION

Credit risk is basically defined as the likelihood that a credit borrower will not manage to fulfill his/her debt obligations in accordance with agreed terms [1]. As a result of the evaluation of the risk, a credit applicant may be qualified and/or classified as creditworthy or not. Many financial organizations like banks and credit card companies carry out risk analysis by employing some credit scoring models in order to rank credit risk and then they develop strategies for loaning accordingly [2]. Conventionally, as a result of the evaluation of the credit risk, low risky borrowers are more likely to get credits whereas high risky ones will be considered as probable insolvent borrowers and usually they will not be granted credit.

1.1. Credit Risk

First step of credit risk analysis is credit scoring. Credit score is a qualitative and indicative metric for credit risk computation that is based on the necessary detailed information given by the credit customers concerning their credit applications [3]. Credit scoring models are utilized by financial institutions for the evaluation of the credit applications [2]. In this regard, credit scoring has several advantages in the sense that it makes customer evaluation easier and quicker. First of all, credit scoring brings the advantage of time savings especially for retail markets where, the number of credit applicants is rather high. By default, a loan officer has to evaluate credit customers one by one spending a lot of time. However credit scoring, systematically evaluates the customers in a short time with much less requirement for face-to-face contacts. What is more is that even less number of working loan officers is sufficient for processing the credit applications in the presence of credit scoring. Secondly, the experts and loan officers evaluating the credit applications can be somehow biased and may not always be sufficiently objective for creditworthiness decision. For example, some features like gender and nationality may be handled in a subjective manner by them. On the contrary, credit scoring is a strictly objective decision making mechanism regardless of gender or nationality of the applicant [2].

Credit risk analysis, as a popular and important financial analysis problem, also has been attracting many researchers' attention. Logistic Regression (LR) and Linear Discriminant Analysis are conventional methods for risk computation and the classification of credit customers. John G. Wiginton, in his paper explains his studies about credit granting decision performance of logit model (Logistic Regression). He also compares logit models with Linear Discriminant Analysis [4, 5]. On the other hand, Bayesian Networks is also studied for credit risk scoring by some researchers as well [6]. B. Baesens and his friends used Bayesian Networks as probabilistic explicit classifiers based on maximum likelihood principle for credit scoring [7]. Support Vector Machines (SVM) is also employed for credit scoring as well as other conventional classification methods like Neural Networks and k-Nearest Neighbors (k-NN) [8 - 12]. There are also some financial studies for credit scoring and risk analysis concerning HMM which is considered as the simplest dynamic Bayesian Network [13-14]. These studies generally deal with number of default risks and prediction of default probable bankruptcies in time horizon.

Credit risk is also emphasized by consultative Basel Accords which deal with guidance and recommendations on banking laws and regulations, released by Basel Committee on Banking Supervision. The Committee, which is in charge of supervising capital requirements for international financial institutions, released The New Basel Capital Accord, namely Basel II, in 2001 with two subsequent revisions in 2002 and 2003 [2]. According to Basel Accords, credit risk is getting more important in the world of finance such that even every ordinary credit applicant firm should be rated by the lenders. This is because the banks are considered to be exposed to profound amount of credit risk which is accepted as one of the major risk components together with market risk and operational risk [11], [15]. Fast growth in consumer credit market also increased the importance of credit risk [16]. In other words, due to the bankruptcy possibility of default of their customers, banks are facing credit risk which should be carefully and seriously analyzed.

Formerly, risk analysis consisted of classifying credit customers in a binary manner as good or bad, representing probable solvent and insolvent borrowers respectively. However, pure classification is not sufficient for modern elaborate risk analysis [2]. Aforementioned Basel Accords also strongly emphasize the importance of computing

credit risks of the customers instead of binary classification as good borrower and bad borrower. One of the most popular risk computing methods is known to be the probability of default (PD) modeling. Probability of default is nothing but a probability concerning the likelihood that credit customer would have a failure in paying his debts on time [17]. More precisely, the lower PD value a credit customer has, the more creditworthy he/she is. In addition, HMM has not been extensively used for PD model building purposes so far.

Many researchers have also been studying about accurate risk computation models [9], [16], [18]. In this regard, in his paper, Alessio A. Saretto declares that a suitable model which utilizes the financial information given for credit application is necessary to be able to have a prediction about the financial picture of the credit borrower in near future. He also insists on the importance of accurate prediction of PD values. Because if a bank assigns higher PD value for a credit packet, then the bank would have to hold more capital which may be somehow redundant and costly [17], [2]. One extension of PD modeling is to determine the risk rating grades indicating the creditworthiness of a borrower firm. Risk rating grades are obtained by mapping of the default probabilities (PD values) into risk rating scales. Many credit risk rating companies like Standard & Poor's and Moody's make use of risk rating grades based on PD values [19-20].

In summary, credit risk analysis consists of classification of credit customers and computation of PD measures as shown in Figure 1.1 below. And this master thesis aims to investigate Hidden Markov Model on PD modeling purposes and classification. For illustrious point of view, in this thesis study, PD modeling ability of HMM is also compared to that of Logistic Regression which is probably the most conventional method of the relevant research area. On the other hand, classification ability of HMM is also compared with that of Logistic Regression and k-Nearest Neighbors.

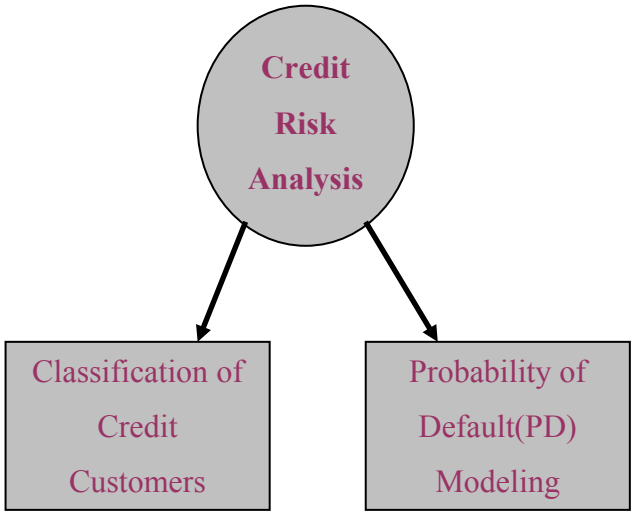


Figure 1.1. Credit Risk Analysis Procedure

1.2. Objectives

This thesis study aims to investigate the performance of Hidden Markov Model (HMM) for credit risk analysis in terms of classification and probability of default (PD) modeling. Here PD modeling assigns default bankruptcy probabilities to credit customers instead of strictly classifying them as good (solvent) borrowers and bad (insolvent) borrowers. The goal is to use HMM for accurate PD modeling and effective classification by utilizing Australian and German Credit Databases both of which are public and freely accessible [21-22]. Results are to be shown in a benchmarking manner with regard to conventional methods like LR and k-NN.

1.3. Organization of the Thesis

Chapter 1 is the introductory part of this thesis. The importance of credit risk, classification of credit applicants and PD modeling is emphasized. Introduction also includes outline and concept motivation about the thesis topic.

Chapter 2 is devoted to scrutinized information and methods about credit classification. The nature and character of the risk analysis is disclosed. Proposed

classification model of HMM is introduced as well as conventional methods like Logistic Regression and k-NN. Last part of chapter 2 shows the results of the experiments in a comparative manner in order to be able to give solid ideas about the performance of newly proposed HMM method.

As for Chapter 3, PD modeling procedure of HMM is explicitly explained with details. For comparison purposes, PD modeling analysis is also performed for Logistic Regression. Last part illustrates the results of newly presented HMM based PD modeling in a benchmarking manner with regard to Logistic Regression based PD modeling.

Chapter 4 is organized as the conclusion part. Summary of the whole study is given comprehensively. Results of the experiments and the performance of HMM are evaluated. Possible future work is also presented in this final chapter.

2. CREDIT CLASSIFICATION

2.1. Credit Data

German Dataset and Australian Dataset are two public access credit databases that this thesis study utilizes [21, 22]. Every row of the databases corresponds to a credit customer with each element of every row being a feature for the customer. More precisely, each column represents an attribute for the customers. In fact an attribute is nothing but a kind of predictive information like credit history, annual income, credit purpose, credit amount and etc… Attributes are used by the loan officers and experts in order to decide whether the credit applicant is creditworthy or not. Both databases have continuous (numerical) and discrete (categorical or symbolic) attributes.

German dataset has 1000 data instances regarding retail credit applications. 300 of the instances are of the bad borrowers and 700 of them are of the good borrowers. German dataset contains a total of 20 attributes 7 of which are numerical and 13 of which are categorical. Some of the attributes presented in German Database are as follows: status of existing checking account (categorical), credit history (categorical), credit purpose (categorical), credit amount (numerical), present employment since (categorical) and etc…

On the other hand Australian database has 690 data instances regarding credit card applications. 383 of the instances belong to insolvent borrowers and 307 of them belong to solvent borrowers, meaning that it is more balanced than German dataset. The dataset has 6 numerical and 8 categorical attributes yielding a total of 14 attributes. However, Australian dataset is a little more cryptic in the sense that the names and descriptions of the attributes are not given explicitly. Instead, the attributes are coded with a numbering format from 1 to 14 such as Attribute1, Attribute2 etc...

Since both databases have numerical and categorical data together, an adjustment for the datasets is necessary in order to make them appropriate for the classifiers of our interest namely, HMM, LR and k-NN. For this reason the numerical data are converted to

categorical format. In other words, continuous values of the attributes are converted to discrete categories according to discrete intervals. For example, the attribute regarding *credit amount*, is a continuous number representing the amount of the loan that the credit applicant wants to borrow. It is denoted as *attribute 4* in German dataset as shown in Table 2.1. Since the credit amount may take a wide range of continuous values, the values of this attribute should be converted to discrete categorical values in accordance with appropriate intervals as follows:

- If the credit amount is smaller than 1000 DM (Deutsche Mark), it is converted to category 1 and encoded as C041 indicating the 1st category of attribute 4.
- Else if the credit amount is between 1000 DM and 4000 DM, it is converted to category 2 and encoded as C042 indicating the 2nd category of attribute 4.
- Else if the credit amount is between 4000 DM and 10000 DM, it is converted to category 3 and encoded as C043 indicating the 3rd category of attribute 4.
- Else if the credit amount is greater than 10000 DM, it is converted to category 4 and encoded as C044 indicating the 4th category of attribute 4.

For both datasets, the number of columns representing input features (attributes) is rather high and it should be decreased to a reasonable number. This should be done intuitively by trial and error method. Alternatively, factor analysis can be used to see which factors are more predictive. Principal component analysis or independent component analysis can also be used for dimension reduction. However in this case components take the place of the factors and it is not very straightforward to see which factors should be used in the model.

In this study, dimension reduction is employed through trial and error method in an intuitive manner. Unused attributes are discarded from both datasets. Finally the number of input features is eliminated to 11 for both of the datasets. Best classification accuracies are achieved with 11 attributes (columns) with HMM according to which we focused on optimizing the performance in this study. In other words, best classification accuracy for Hidden Markov Model is obtained using appropriate combination of 11 input features for both datasets. For comparison purposes, exactly the same reduced data are used for other

classifiers: Logistic Regression and k-Nearest Neighbors. From here on, whenever the dataset is mentioned, the final reduced datasets are referred.

After dimension reduction is finalized, the characteristics of the datasets have arisen as in the following Table 2.1 and Table 2.2. The total number of all categories of all used attributes is 45 for German dataset and 57 for Australian dataset.

After the above adjustments are finished the datasets are exposed to classification experiments by using HMM, LR and k-NN.

Table 2.1. The Character of German Dataset

Employed Features	**Description of Attributes (Feature)**	**# of categories for attribute**	**Encoding of attributes**
Feature 1	Status of Existing Checking Account	4	C011, C012, C013, C014
Feature 2	Duration in month	4	C021, C022, C023, C024
Feature 3	Credit History	5	C031, C032, C033, C034, C035
Feature 4	Credit Amount	4	C041, C042, C043, C044
Feature 5	Savings accounts/bonds	5	C051, C052, C053, C054, C055
Feature 6	Present employment since	5	C061, C062, C063, C064, C065
Feature 7	Installment Rate	4	C071, C072, C073, C074
Feature 8	Other debtors/Guarantors	3	C081, C082, C083
Feature 9	Property	4	C091, C092, C093, C094
Feature 10	Other Installment Plans	3	C101, C102, C103
Feature 11	Job	4	C111, C112, C113, C114

Table 2.2. The Character of Australian Dataset

Employed Features	Description of Attributes (Feature)	# of categories for attribute	Encoding of attributes
Feature 1	Attribute 1	2	C011, C012
Feature 2	Attribute 2	9	C021, C022, C023, C024, C025, C026, C027, C028, C029
Feature 3	Attribute 3	5	C031, C032, C033, C034, C035
Feature 4	Attribute 5	14	C041, C042, C043, C044, C045, C046, C047, C048, C049, C0410, C0411, C0412, C0413, C0414
Feature 5	Attribute 6	9	C051, C052, C053, C054, C055, C056, C057, C058, C059
Feature 6	Attribute 7	6	C061, C062, C063, C064, C065, C066
Feature 7	Attribute 8	2	C071, C072
Feature 8	Attribute 9	2	C081, C082
Feature 9	Attribute 10	3	C091, C092, C093
Feature 10	Attribute 11	2	C101, C102
Feature 11	Attribute 12	3	C111, C112, C113

2.2. Classification with Hidden Markov Model

One of the main objectives of this thesis study is to see whether HMM can be effectively used for credit data classification. For this reason, this section introduces Hidden Markov Model and discloses how it can be utilized for classifying credit data. As mentioned in the introductory chapter, many methods like Logistic Regression, Support Vector Machines, Linear Discriminant Analysis, Neural Networks, k Nearest Neighbors have attracted many researchers' attention [4, 5], [8-12]. Bayesian Networks also have

been used for credit classification [6, 7]. However, as the simplest form of dynamic Bayesian Networks, HMM has not been studied for credit customer classification and credit risk analysis extensively.

2.2.1. Hidden Markov Model

Hidden Markov Model can be considered as a statistical model based on a Markov process with unknown parameters influencing observable outputs. Unlike a regular Markov model, the states are not visible for a HMM. Since regular Markov models are not very well fitting the real life problems they are extended to hidden Markov models so that the observations become probabilistic influences of states which are hidden. Because of its versatile and rich mathematical background, the main action with hidden Markov models generally have arisen as finding out the hidden parameters which may be utilized to employ pattern recognition applications like speech and image recognition and bioinformatics [23, 24]. This thesis aims to deduce how to apply HMM on credit risk analysis.

Hidden Markov models were first introduced in late 1960s by L.E. Baum and his friends with a series of papers [25, 26]. In 1970s HMM was first used in speech recognition. The importance of HMM on pattern recognition was however noticed in 1980s. In this regard, Lawrence R. Rabiner, published his famous illustrative paper in which theoretical aspects of statistical Markov models were reviewed in details [24]. This paper is also utilized in developing our model for this thesis study.

The basis of hidden Markov model starts with discrete Markov process. Consider a system in which one of the N states of the system is emitted at equally spaced time intervals, i.e. t=1, 2, 3, … The state changes are assumed to occur according to state transition probabilities associated with the model. For the case N=3, the system model is shown in Figure 2.1.

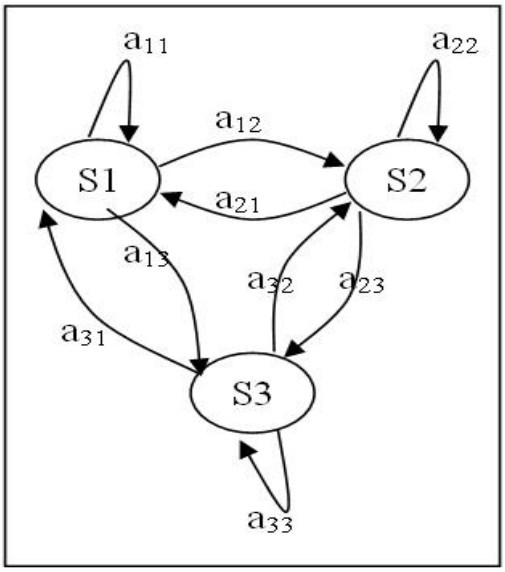


Figure 2.1. Markov Chain with three states and corresponding transitions

According to Markov process the system is assumed to be in one of the N distinct states (S_1, S_2, ... , S_N) at any time instant t=1, 2, 3, … where q_t is the actual state at time t. For a first order Markov chain, probabilistic relation between the states is given as:

$$P\{q_t = S_j \setminus \mathrm{q_{t-1}} = S_i, q_{t-2} = S_k, \ldots\} = P\{q_t = S_j \setminus \mathrm{q_{t-1}} = S_i\} \tag{2.1}$$

Equation (2.1) shows that state at time t, q_t, only depends on the state at time t-1, q_{t-1}. State transition probabilities denoted by a_{ij}, are expressed by the following equation:

$$a_{ij} = P\{q_t = S_j \setminus \mathrm{q_{t-1}} = S_i\} \tag{2.2}$$

with $1 \leq i, j \leq N$ and $a_{ij} \geq 0$ provided that:

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{2.3}$$

The system described above represents an observable Markov process since each state at any instant is directly related to a physical observable event. However, extension from regular Markov model to hidden Markov model lies on the fact that the observations are probabilistic functions of the states [24]. This means that there is no direct access to

states, but the observations, influenced by the hidden states, are visible for Hidden Markov Model as shown in Figure 2.2.

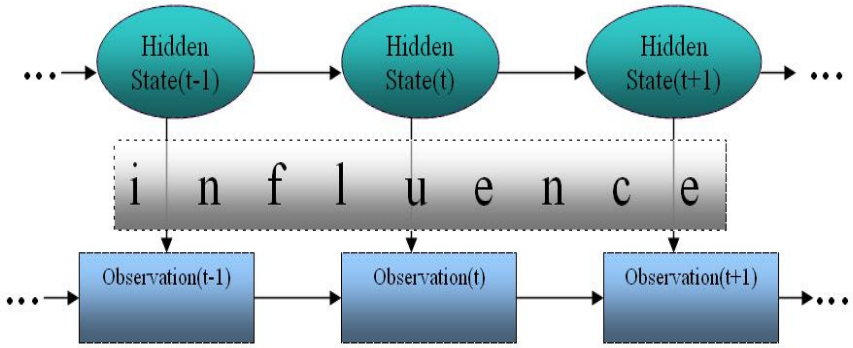


Figure 2.2. Observations influenced by hidden states in a HMM structure

Therefore, a HMM is described and identified by 5 characteristics:

- The first one is the number of hidden states in the model which is denoted by N. The states are denoted as S_1, S_2, …, S_N.

- The second characteristic is the number of distinct observation symbols, M, for any state. M is, in fact, the total number of possible observation symbols that a hidden Markov model can emit in any hidden state. Any one of the M observation symbols may be emitted at each sampling time of the discrete process according to the hidden state in which the system is in. The observation symbols can be denoted as v_1, v_2, …, v_N.

- The third character is the state transition probabilities: a_{ij} such that it satisfies equations (2.2) and (2.3).

- The fourth character is observation symbol probabilities in state i, $b_i(k)$, where

$$b_i(k) = P\{v_k @ t \setminus q_t = S_i\} \tag{2.4}$$

with the condition $1 \leq i \leq N$ and $1 \leq k \leq M$

- And the fifth character of the model is the initial state probabilities π_i, where

$$\pi_i = P\{q_1 = S_i\}, \tag{2.5}$$

and where $1 \leq i \leq N$.

Provided that these 5 characters are specified, then HMM can generate an observation sequence accordingly, such as $O=O_1O_2\bullet\bullet\bullet O_T$ where each O_t is an observation symbol at time t and T being the total number of observations [24].

Hidden Markov Models are generally utilized for three kind of problems so far.

First problem is questioning the computation of the probability of an observation sequence with the given model. More precisely one must answer the question: "What is the probability that the given model generates a specific observation sequence?". In other words, as a second point of view, the problem asks how much a model fits for a given sequence. This problem can be solved by the so called forward backward algorithm [24].

Second problem deals with determining the best state sequence regarding the given observation sequence [24]. Viterbi algorithm solves this problem with maximization of arguments (argmax) approach [27, 28].

Third problem, the most difficult one, is also known as the learning problem. The problem focuses on how to select the best model parameters so as to maximize the probability that the observation sequence is generated by the learnt model [24]. This problem is overcome by Baum-Welch algorithm based on expectation maximization method [29]. Conventionally, Baum Welch algorithm tries to optimize the model parameters so that the likelihood of a sequence about belonging to the trained model is maximized. Generally, the logarithm of the likelihood (log-likelihood) is used as a measure about this problem.

2.2.2. Proposed Method for HMM Classifier

Since credit data have no time concept, there existed a problem about how to fit HMM for classification of such a credit data which does not run in time. What the dataset has, is categorical values (shortly categories) representing the attributes on every row for each credit customer. For German dataset, for example, the first element of every row is one of the 4 categorical variables corresponding to first attribute, *Status of Existing Checking Account* as seen in Table 2.1 above. Since this attribute has 4 categories, first column of every row (credit customer) is either of C011, C012, C013 and C014. Likewise second column of every row is either of C021, C022, C023 and C024 and so on…

In order to introduce a time concept to our system, each element of every row (representing a categorical value for the corresponding attribute) is considered as an observation symbol for Hidden Markov Model. According to this point of view, every row corresponding to a credit customer in the dataset becomes an observation sequence for the model. Therefore, since we utilize 11 attributes on our datasets, each observation sequence has a length of 11, meaning that we have 11 nodes. Therefore our proposed model has one initial state as usual and 10 more state transitions.

What is worthy of noticing in this model, is that for example for the German dataset, (as an observation symbol) first node can emit only either one of the categories of attribute 1, namely C011, C012, C013 and C014. Similarly, second node (second element of the row), emits either of C021, C022, C023 and C024 which are the categories of attribute 2 and so on... Hence, in the proposed model, there is no probability that a category of a certain attribute, can exist at the node of another attribute as an observation symbol. Therefore an observation symbol is never repeated in an observation sequence. Nodes associated with relevant attributes always emits categories of the corresponding attribute. The first node emits one of the categories belonging to attribute 1, the second node emits one of the categories belonging to attribute 2, and so on… Figure 2.3 makes the proposed model more clear for better understanding.

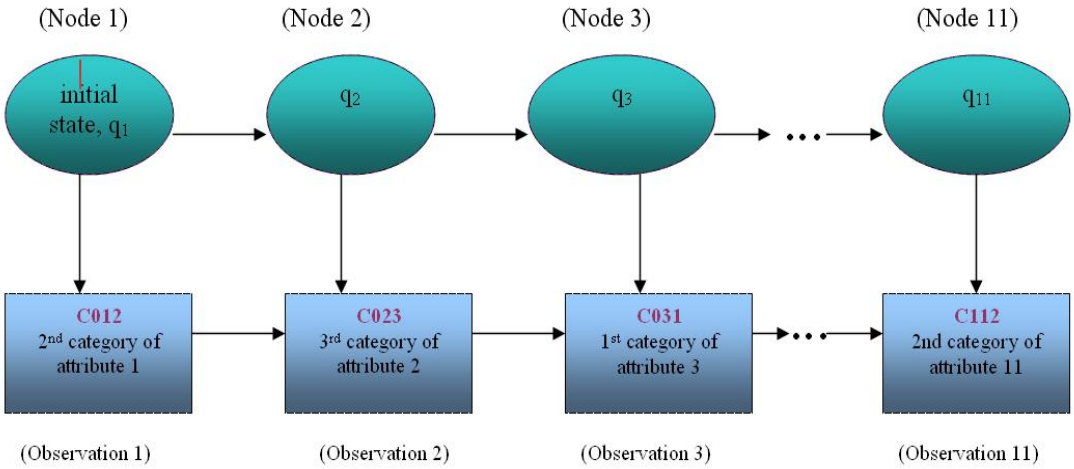


Figure 2.3. Illustration of observation and state sequence for proposed model

In the figure, the values of observation symbols like C012, C023, C031 are given for the purpose of illustration. For example, the first observation should be C011, C013 or C014 as well as C012 however it cannot have a different value other than these since the first node only emits the categories of the first attribute. It is also worth noticing that the total number of observation symbols in the model is the sum of all possible categories of all attributes. For German dataset, 11 attributes yield a total of 45 observation symbols whereas 11 attributes of Australian dataset have a total of 57 observation symbols. The total number of observation symbols can also be found if the values in the third column (number of categories for attributes) of Table 2.1 and Table 2.2 is summed up respectively. In conclusion, lack of time concept in credit data system is overcome by using categories of the attributes as nodes of the Hidden Markov Model.

Matlab's HMM toolbox (by Kevin Murphy) is utilized for HMM analysis in this thesis study. According to proposed model, the user enters the number of hidden states to the system where the optimum number of hidden states are determined according to consecutive experiments in a trial and error approach. In addition to number of hidden states, the total number of observation symbols described in previous paragraph is another parameter to be specified for the system.

2.2.3. Implementation of Proposed Method

After determining the structure of the employed model, implementation of the model has come into picture. The first step of the implementation is training as usual. However our proposed implementation algorithm is based on training of two models instead of one. These two models are named as the good model and bad model. Good model is trained using the data belonging to good (solvent) credit borrowers whereas bad model is trained using the data belonging to bad (insolvent) credit borrowers.

Training set contains a total of 500 observation sequences 250 of which are of good borrowers and remaining 250 of which are of bad borrowers. Test sets, however on the other hand, contains 100 observations of equally weighted between good and bad borrowers meaning that 50 of the 100 observations belong to good borrowers' data and remaining 50 are of bad borrowers. 6-fold cross validation is employed for the computations. So there are 6 subsets of data with each subset consisting of 100 equally weighted data.

Implementation algorithm starts with random initialization of good model and bad model. After the user inputs number of hidden states and total number of all possible observation symbols; initial state vectors, observation symbol probability distribution matrices and hidden state transition matrices are automatically initialized for both models according to proposed algorithm. After random model initializations, a training process is employed for both premature models. Training process, undoubtedly, is carried on by Baum-Welch learning algorithm based on expectation maximization procedure [25, 26].

Once the randomly initialized models are trained by Baum Welch algorithm, the next step is to find which model (good model or bad model) is more likely to generate an observation sequence in the test sample. In other words, we compute the log-likelihood that each test sequence belongs to both models. Hence this step is finished with two log-likelihood values regarding good model and bad model.

Next step is to determine the class of the test sequence by comparing the log-likelihood values as follows:

- If log-likelihood value associated with good model (good LL) is bigger than the log-likelihood value associated with bad model (bad LL), then this test sequence (credit applicant) is classified as good meaning that he is creditworthy.

- If good LL is less than bad LL, then this test sequence is classified as a bad applicant and he is not granted credit.

Therefore our procedure starts with fully random initializations of reference models as good model and bad model. Procedure continues with training good model and bad model via expectation maximization procedure where good model is trained with training samples belonging to good borrowers and bad model is trained with training samples belonging to bad borrowers. Final step is to determine which model is more likely to generate the test sequence by computing log-likelihood values associated with two models. The algorithm flow-chart can be seen on Figure 2.4

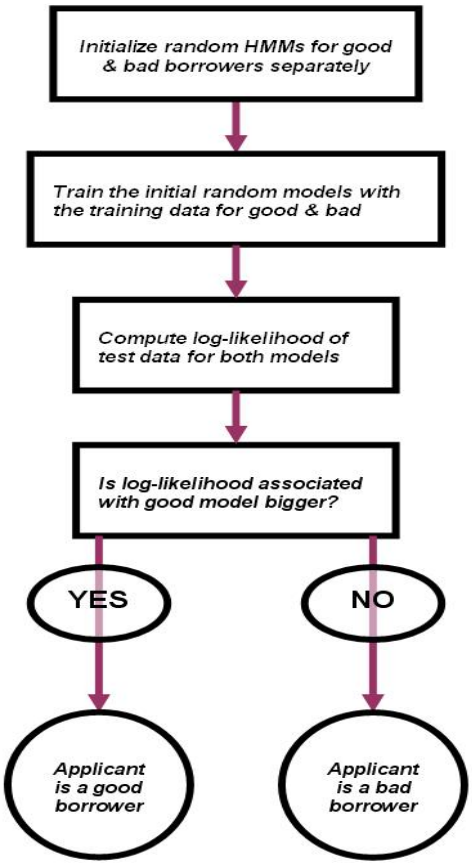


Figure 2.4. HMM Classification Algorithm

However since our reference models (good and bad models) are trained from random initializations, we have to implement the algorithm on Figure 2.4 a few times to get more reliable and accurate results. Final classification decision is made according to majority rule after repetitions of classification algorithm shown in Figure 2.4. So it is a good idea to repeat the procedure for an odd number of times and then decide the class of the test sequence by majority rule. In our case 15 repetitions are said to be sufficient for consistency of classification. More repetitions are supposed to bring more consistency and accuracy to classification of credit customers. On the other hand, more repetitions also bring more computational costs.

In the early stages of the study, the proposed algorithm did not give sufficiently satisfactory results. That's why we had to tune some parameters of the system. Parameter adjustments and model tuning are done by trial and error with the first subset of six-fold cross validated data as shown in Table 2.3 below. Adjustable parameters are said to be the number of repetitions of classification algorithm, the number of hidden states, threshold for the slope of log-likelihood in iterations, number of training data and maximum number of iterations as follows:

- As one of the most influential parameter of the system, the optimum number of hidden states is determined as 15 by trial and error approach for both datasets. If we use less number of hidden states, then classification accuracy and consistency become lower. However if we determine the number of hidden states to 15 in the model, we noticed that we have reached maximum accuracy for all cross validated data. Increasing the number of hidden states more than 15 did not bring more improvement in accuracy or in consistency. In other words, whenever we run the algorithm with 15 or more hidden states, it gives the same accuracy in every trial. As seen in the first 3 rows of Table 2.3 below, using 15 hidden states resulted in a classification accuracy of 76 per cent where using 20 hidden states actually did change nothing in means of classification accuracy on the condition that other parameters are kept constant. On the other hand using less number of hidden states, say 11, resulted in an accuracy of 74.2 per cent. It is also worth emphasizing that the data in Table 2.3 belong to first subset of German dataset in 6-fold cross validation scheme.

- Attributes employed in the model, have also significant influence on the classification accuracy and consistency. As mentioned in Section 2.1 only specified 11 of the attributes are utilized in the system because our proposed model has given the best results with 11 attributes in both datasets. Using less attributes or more attributes resulted in lower classification accuracy as seen in the last two rows of Table 2.3 below. When two attributes are added to the system, yielding a total of 13, then the accuracy decreased from 76 per cent to 73 per cent.

- Number of training sequences is also affecting the results severely. In the beginning of the thesis study, a total of 200 training samples were used. However, increasing this number to 500 led to much better results noticed in fourth and fifth rows of Table 2.3 below.

Table 2.3 HMM Classification scheme for first subset of German dataset

Experiment number	**# of attributes**	**# of training data**	**# of test data**	**# of hidden states**	**Average Good Borrower Accuracy (%)**	**Average Bad Borrower Accuracy (%)**	**Average Total Accuracy (%)**
1	11	500	100	11	79.2	69.2	**74.2**
2	11	500	100	15	76	76	**76**
3	11	500	100	20	76	76	**76**
4	8	200	100	3	62.4	61.2	**61.8**
5	8	500	100	3	78	62.8	**70.4**
6	11	500	100	15	76	76	**76**
7	13	500	100	15	76	70	**73**

It is also worth mentioning that threshold for the slope of log-likelihood in iterations is by default 10^{-4} on Matlab's HMM toolbox. However this number is employed as 5×10^{-4} for better convergence of the iterations in all experiments. As we decrease this threshold, classification accuracy is supposed to increase in parallel to computational cost.

Maximum number of iterations is another parameter for the termination of iterations employed in expectation maximization algorithm. Its default value in MATLAB is set to

15. However we did not let the iterations terminate due to reaching maximum number of iterations. The iterations that terminated due to reaching maximum limit are cancelled and a new iteration is immediately started until the iteration is terminated because of converging meaning that the convergence threshold is reached.

In the following tables, Table 2.4 and Table 2.5, the detailed results of HMM classification procedure are given for German and Australian Datasets respectively. Classification performances for all 6 subsets are given in details.

Table 2.4. Results of HMM experiments on German Dataset

Fold (subset) number	# of attributes	# of training data	# of test data	# of hidden states	Good Borrower Accuracy (%)	Bad Borrower Accuracy (%)	Average Total Accuracy (%)
1	11	500	100	15	76	76	**76**
2	11	500	100	15	74	82	**78**
3	11	500	100	15	66	74	**70**
4	11	500	100	15	72	62	**67**
5	11	500	100	3	58	72	**65**
6	11	500	100	15	80	68	**74**
OVERALL AVERAGES OF SIX SUBSETS					71	72.33	**71.67**

Table 2.5. Results of HMM experiments on Australian Dataset

Fold (subset) number	# of attributes	# of training data	# of test data	# of hidden states	Good Borrower Accuracy (%)	Bad Borrower Accuracy (%)	Average Total Accuracy (%)
1	11	500	100	15	88	78	**83**
2	11	500	100	15	86	84	**85**
3	11	500	100	15	88	84	**86**
4	11	500	100	15	80	88	**84**
5	11	500	100	15	88	86	**87**
6	11	500	100	15	86	82	**84**
OVERALL AVERAGES OF SIX SUBSETS					86	83.67	**84.83**

In Table 2.4 and Table 2.5, classification performance of each subset of the 6-fold cross validated data is shown. 'Average good borrower accuracy', 'average bad borrower accuracy' and 'average total accuracy' are found by the mathematical averages of corresponding accuracy values of the 6 subsets presented in the above tables. According to HMM results, 86 per cent of average good borrower accuracy, 83.67 per cent of average bad borrower accuracy and 84.33 per cent of average total accuracy is obtained for Australian Dataset whereas 71 per cent of average good borrower accuracy, 72.33 per cent of average bad borrower accuracy and 71.67 per cent of average total accuracy is obtained for German Dataset.

The order of attributes employed in the model is also affecting the accuracies. More precisely if we change the order of the input features, then the classification accuracies may improve or may get worse. For example if we arrange the order of the features in the following order: (*attribute 4, attribute 8, attribute 5, attribute 1, attribute 3, attribute 9, attribute 11, attribute 7, attribute 10, attribute 2, attribute 6*) instead of default regular ordering, the results happen to come out as in Table 2.6 and Table 2.7 below for German and Australian datasets respectively. One can see that the accuracies have changed a bit with respect to default ordering results seen in Table 2.4 and in Table 2.5 above. It is also worth noticing that there are 11! combinations for the order of input features in the model. Therefore it is not very straightforward to determine the best ordering without tyring out these 11! different orderings.

Table 2.6. Results of HMM experiments on German Dataset for different ordering

Fold (subset) number	**# of attributes**	**# of training data**	**# of test data**	**# of hidden states**	**Good Borrower Accuracy (%)**	**Bad Borrower Accuracy (%)**	**Average Total Accuracy (%)**
1	11	500	100	15	74	76	**75**
2	11	500	100	15	62	82	**72**
3	11	500	100	15	58	74	**66**
4	11	500	100	15	72	62	**67**
5	11	500	100	3	76	64	**70**
6	11	500	100	15	56	70	**63**
OVERALL AVERAGES OF SIX SUBSETS					66.33	71.33	**68.83**

Table 2.7. Results of HMM experiments on Australian Dataset for different ordering

Fold (subset) number	# of attributes	# of training data	# of test data	# of hidden states	Good Borrower Accuracy (%)	Bad Borrower Accuracy (%)	Average Total Accuracy (%)
1	11	500	100	15	80	70	**75**
2	11	500	100	15	84	84	**84**
3	11	500	100	15	86	82	**84**
4	11	500	100	15	68	86	**77**
5	11	500	100	15	88	86	**87**
6	11	500	100	15	84	82	**83**
OVERALL AVERAGES OF SIX SUBSETS					81.67	81.67	**81.67**

As seen from Table 2.4 and Table 2.6, the overall accuracy is changed from 71.67 per cent to 68.83 per cent for German dataset according to six-fold cross validation procedure. The change is also seen in Australian dataset in the case of a different input feature arrangement. On Table 2.7, one can see that the overall accuracy have arisen as 81.67 per cent for the aforementioned different ordering instead of 84.83 per cent for default regular ordering shown in Table 2.5.

For illustrative purpose of transition characteristics, the transition matrices of both models are shown in the following tables, Table 2.8 and Table 2.9, for the third subset of German dataset. The training data consist of first, second, fourth, fifth and the sixth subsets where the test data consist of the third subset. It is also worth noticing that the transition matrices may vary from iteration to iteration since the initial transition matrices are chosen fully randomly. As seen, some states are not met at all; i.e. *state 5* of good model and *state 8* of bad model. And also, some states have absolute transitions (transition probability is equal to 1) to another state.

Table 2.8. Transition Matrix of 'good model' for third subset of German Dataset

	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10	State 11	State 12	State 13	State 14	State 15
State 1	0	0	0	0.8941	0	0	0	0.1059	0	0	0	0	0	0	0
State 2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
State 3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
State 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
State 5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
State 6	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
State 7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
State 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
State 9	0.0161	0	0	0	0	0	0	0	0	0	0.9839	0	0	0	0
State 10	0	0	0	0	0	0	0	0	0	0	0	0.0661	0	0	0
State 11	0	0	0	0.4772	0	0	0	0.5228	0	0	0	0	0	0	0
State 12	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
State 13	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
State 14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
State 15	0	0.8035	0.0478	0	0	0	0.1486	0	0	0	0	0	0	0	0

Table 2.9. Transition Matrix of 'bad model' for third subset of German Dataset

	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10	State 11	State 12	State 13	State 14	State 15
State 1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
State 2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
State 3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
State 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
State 5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
State 6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
State 7	0	0	0	0	0.0222	0	0	0	0	0	0	0	0	0.9778	0
State 8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
State 9	0	0	0	0	0	0.309	0	0	0	0.2889	0.4021	0	0	0	0
State 10	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
State 11	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
State 12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
State 13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
State 14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
State 15	0	0	0.9747	0	0	0	0	0	0	0	0	0.0253	0	0	0

2.3. Classification with Logistic Regression

Logistic Regression models are generally employed for computing the probability of occurence of a certain event by making use of total contribution of predictive input variables. Logistic Regression is widely used in medical and social sciences such as predicting a disease on a patient or tendency of people to buy a specific product [30]. As one of the most popular and conventional methods in credit scoring, Logistic Regression (LR) can be used for classification of credit customers as well as predicting bankruptcy probability [4], [5]. The general principle of Logistic Regression is that total contribution of all employed attributes is appropriately transformed into probability of insolvency. A threshold on this insolvency probability is then employed to distinguish between good and bad borrowers.

2.3.1. Logistic Regression Model

Logistic Regression simply uses the method of transforming input vector into probability of occurrence of a certain event. The input vector consists of variables that are partially employed in determining the probability of occurrence. The transformation is carried out by equation (2.6) which is also called the logistic function.

$$f(z) = \frac{1}{1+e^{-z}} \tag{2.6}$$

In this equation, z is the input vector and f(z) is the output. The challenge is to fit the input to a logistic curve as shown in Figure 2.5.

In equation (2.6) the input, *z*, can be in the range (-∞,+∞) and the output is in the range (0, 1) so that the transformation corresponds to a probability value concerning the predictive input variables.

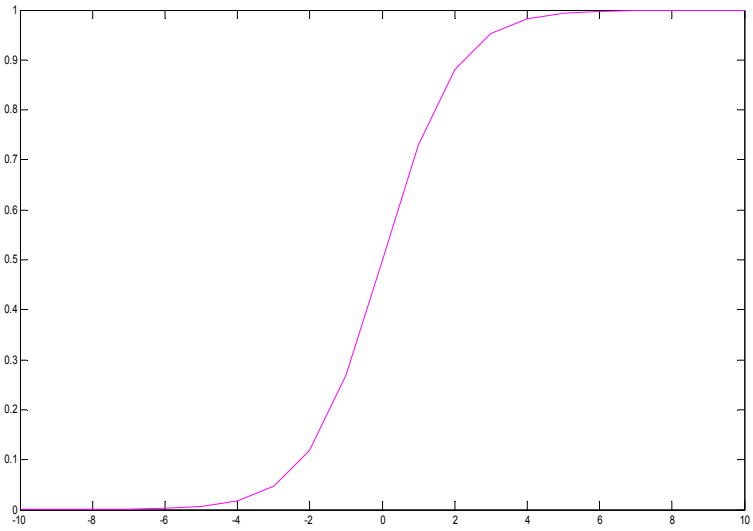


Figure 2.5. Logistic Curve

The input variable z can be expressed as in equation (2.7).

$$z = \beta_0 + \beta_1 \chi_1 + \beta_2 \chi_2 + \ldots + \beta_k \chi_k \tag{2.7}$$

In equation (2.7) βo is called the intercept, and β_1, β_2, β_3 and so on, are regression coefficients of x_1, x_2, x_3 and so on, respectively where x_1, x_2, x_3 and so on, are the predictive input variables. They are also called risk factors. The regression coefficients somehow represent the importance and effect of the corresponding risk factors on the probability output. The bigger the coefficient is, the more influential the risk factor is on the output [30]. Therefore input variable *z* is a metric concerning the total contribution of all risk factors employed in the model.

In the literature, logistic regression is also expressed with logit function as in equation (2.8) below.

$$\ln\left(\frac{f(z)}{1-f(z)}\right) = \text{logit}(f(z)) = \beta_0 + \beta_1 \chi_1 + \beta_2 \chi_3 + \ldots \tag{2.8}$$

According to equation (2.8), logarithm of the ratio of a probability of an event to the probability of complementary event is related to the weighted sum of all risk factors.

Generally, *f(z)*, the probability output, is used for binary classification for the corresponding input variables. The class of the input is determined according to a pre-determined threshold for the ouput. If the output has a bigger value than the the threshold then the corresponding input is counted as belonging to one class and vice versa [31].

2.3.2. Classification of Credit Customers with Logistic Regression

In credit classification case, every attribute corresponds to a risk factor employed in logistic regression. Values of every attribute are taken into regression process for a user. The data format held in logistic regression is exactly the same as Hidden Markov Model case. There are eleven attributes used in the model just like HMM case. Therefore we have an input space of dimension 11 for both datasets. For example by looking at Table 2.1, German dataset has 4 different categorical values for the first dimension corresponding to first attribute. These 4 values are encoded as 1, 2, 3 and 4 respectively and directly employed in Logistic Regression computations. As for second dimension corresponding to second attribute in German dataset, as seen in Table 2.1, there are again 4 values which are again encoded as 1, 2, 3 and 4 respectively. Therefore, each user's data consists of a row of dimension 11 where every element of this row is a categorical output for the corresponding feature. Hence, this row is nothing but a coordinate of a user's data point in a space of dimension 11.

Logistic Regression algorithm takes the row vectors belonging to training data as input to the system. After regression process of the training data, Logistic Regression algorithm determines the coefficients of every 11 attributes (factors). Once the coefficients (β_0, β_1,…, β_{11}) including the intercept term are determined, next step is to compute the output, *f(z)*, of the test data to see what score the test data have with those coefficients. *f(z)* is computed by equation (2.6). It is also worth noticing that β_0 is the intercept term, β_1 is the coefficient corresponding to first attribute, β_2 is the coefficient corresponding to second attribute, and so on.

After all the outputs (*f(z)*) corresponding to each test data are computed, next step is to classify the test data by using an appropriate threshold [32]. In our case the best threshold has come out to be 0.3775 ($\approx$1-[1/(1+$e^{-0.5}$)]). So, test data having outputs (bankruptcy probabilities indeed) greater than this threshold are classified as insolvent applicants meaning that it is dangerous to grant credit to such credit applicants. On the other hand, test data having outputs less than the threshold are counted as solvent borrowers and they can be granted credit. The procedure is summarized in Figure 2.6.

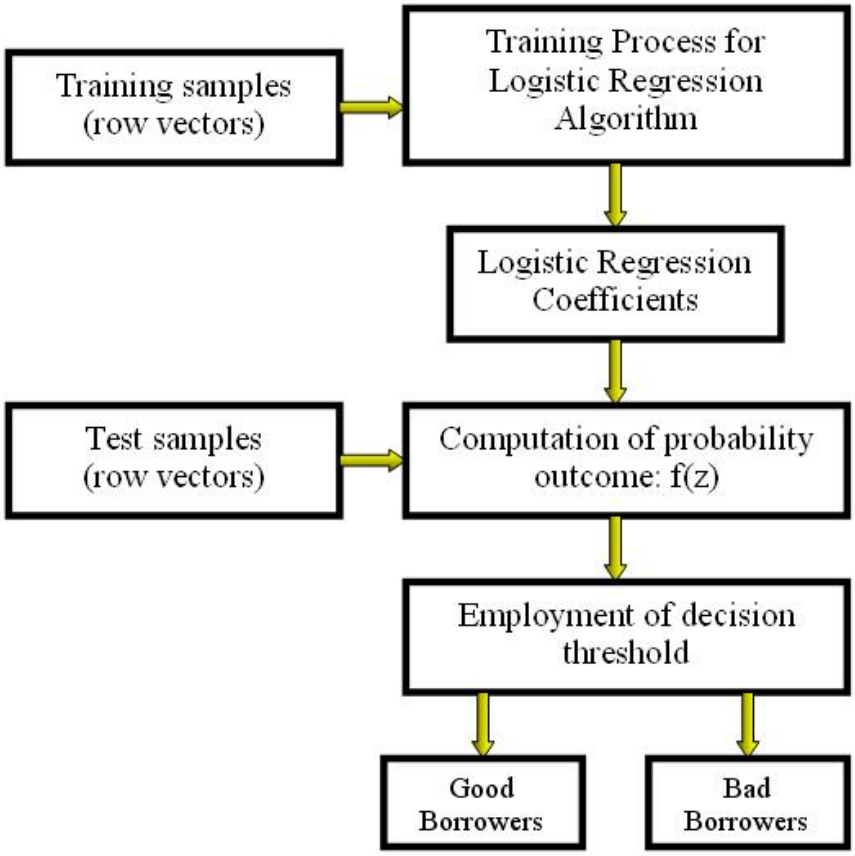


Figure 2.6. Classification process of Logistic Regression

Just like the case in HMM, since we have utilized equal number of training and test data from both classes, there are 250 training samples belonging to solvent borrowers and 250 training samples belonging to insolvent borrowers. Similarly 50 test samples are of good borrowers whereas the remaining 50 test samples are of bad borrowers. In the experiments 6-fold cross validation is employed. LR computations are done via web based LR calculator [33].

In the following tables, Table 2.10 and Table 2.11, the detailed results of LR classification process are given for German and Australian Datasets respectively. Classification performances for all 6 subsets are given in details.

Table 2.10. Results of Logistic Regression experiments on German Dataset

Fold (subset) number	Classification Threshold	# of training data	# of test data	Good Borrower Accuracy (%)	Bad Borrower Accuracy (%)	Average Total Accuracy (%)
1	0.3775	500	100	68	82	**75**
2	0.3775	500	100	46	92	**69**
3	0.3775	500	100	38	80	**59**
4	0.3775	500	100	60	82	**71**
5	0.3775	500	100	66	74	**70**
6	0.3775	500	100	46	86	**66**
OVERALL AVERAGES OF SIX SUBSETS				54	82.67	**68.33**

Table 2.11. Results of Logistic Regression experiments on Australian Dataset

Fold (subset) number	Classification Threshold	# of training data	# of test data	Good Borrower Accuracy (%)	Bad Borrower Accuracy (%)	Average Total Accuracy (%)
1	0.3775	500	100	82	80	**81**
2	0.3775	500	100	90	70	**80**
3	0.3775	500	100	90	86	**88**
4	0.3775	500	100	84	90	**87**
5	0.3775	500	100	90	90	**90**
6	0.3775	500	100	82	86	**84**
OVERALL AVERAGES OF SIX SUBSETS				**86.33**	**83.67**	**85**

In Table 2.10 and Table 2.11, classification performances of each subset of the 6-fold cross validated data are shown. ‘Average good borrower accuracy’, ‘average bad borrower accuracy’ and ‘average total accuracy’ are found with the mathematical averages of the corresponding accuracy values of the 6 subsets presented in the tables. According to Logistic Regression procedure results, 86.33 per cent of average good borrower accuracy, 83.67 per cent of average bad borrower accuracy and 85 per cent of average total accuracy

is obtained for Australian Dataset whereas 54 per cent of average good borrower accuracy, 82.67 per cent of average bad borrower accuracy and 68.33 per cent of average total accuracy is obtained for German Dataset.

2.4. Classification with k Nearest Neighbors

k Nearest Neigbors (k-NN) classifier can also be used for classifying credit data by making use of k number of closest training samples in the feature space [34]. k-Nearest Neigbors, which is a common technic in pattern recognition and non-parametric statistics, has also been studied for credit customer classification [10]. For example, W. E. Henley and D. J. Hand, in their paper, insist on the three advantages of k-Nearest Neighbors in credit risk modeling. According to them, first advantage of k-NN is that k-NN can handle irregularities in risk function concerning the input feature space. Secondly, for multi-dimensional data, k-NN is more likely to have better performance. Third advantage is that k-NN is an intuitive and easily perceived method for business people [10].

2.4.1. k - Nearest Neighbors Model

k- Nearest Neighbors algorithm, simplest of all machine learning algorithms, utilizes k number of nearest neighboring training sample points to determine the class of the test instance where k is an integer. The class of the test sample is determined according to the class of the majority of these k nearest training samples. In binary classification problems, it is very useful to take k as an odd integer for easier decision.

In literature, the concept of being nearest is measured with some different metrics including Euclidean distance, Hamming distance, Manhattan distance etc… In our study, for credit customer classification problem, Euclidean distance is utilized as in eqation (2.9)

$$d(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad (2.9)$$

where d(P,Q) is the n-space Euclidean distance between two points P and Q represented as P=($p_1, p_2, ..., p_n$) and Q=($q_1, q_2, ..., q_n$). In our case, n=11, corresponding to 11 attributes introduced in our model.

First step of this algorithm starts with computation of the distances between all training data and the test sample. After all the distances with respect to the test sample are computed, the values of the distances are to be sorted in order to select smallest k of them. These k smallest distances correspond to k nearest training samples with respect to the test sample. Next step involves evaluating the classes of these k nearest neighbors. The majority class belonging to k nearest neighbors is assigned as class of the test point. In other words, more frequent class is determined as the class of the test sample [34]. The algorithm is explained in Figure 2.7 below.



Figure 2.7 k-Nearest Neighbors Algorithm

The value of k, the number of nearest neighbors, is a parameter for improving the performance of k-NN algorithm because k-NN algorithm is sensitive to local character of the data. Using a larger k will result in more noise avoidance. On the other hand, the boundaries between classes may become more ambigouos when k is big [34]. In our study the best k is determined by trial and error approach. Since all the distances associated with a test sample is calculated for every training instance, computational cost of k-NN is supposed to be rather much.

2.4.2. Classification of Credit Customers with k-NN

Data structure for k-NN algorithm is taken exactly the same as the Logistic Regression work. Therefore, each unit of the data is again in the form of row vectors. Each row vector represents a credit applicant where each element of the rows corresponds to relevant attribute concerning the credit applicant. All the elements are consisting of discrete integers representing categorical values for the attribute. Each row has a length of 11 which means that 11 attributes are employed in our model as in previous cases. For example by looking at Table 2.2, Australian dataset has 2 different categorical values for the first dimension corresponding to first attribute. These 2 values are encoded as 1 and 2 respectively and directly employed in k-NN computations. As for second dimension corresponding to second attribute in Australian dataset, as seen in Table 2.2, there are 9 values which are again encoded as consecutive integers such as 1, 2, 3, 4, 5, 6, 7, 8 and 9 respectively.Therefore, each row, in fact, represents a coordinate of a point in 11 dimensional space.

In our credit customer classification problem, k-NN algorithm has given the best accuracy with 19 nearest neighbors (k=19) for both datasets. 6-fold cross validation is used in k-NN as in previous cases. So if more than 9 nearest training samples belong to good borrowers, then the test sample is classified as solvent borrower according to majority voting of 19 nearest neighbors. On the other hand if less than 10 of these 19 training samples belong to good borrowers, then the borrower is considered as a probable insolvent borrower and she/he may have a bankruptcy in near future. It is also worth saying that k-NN algorithm computations are held with specific Matlab code where 250 training data

points belong to good borrowers and 250 training data points belong to bad borrowers, yielding a total of 500 training data. Likewise, a total of 100 test data points are used. 50 of them are of solvent borrowers and remaining 50 of them are of insolvent borrowers.

In the following tables, Table 2.12 and Table 2.13, the detailed results of k-Nearest Neighbors classification process are given for German and Australian Datasets respectively. Classification performances for all 6 subsets are given in details.

Table 2.12. Results of k-Nearest Neighbors experiments on German Dataset

Fold (subset) number	k (number of neighbors)	# of training data	# of test data	Good Borrower Accuracy (%)	Bad Borrower Accuracy (%)	Average Total Accuracy (%)
1	19	500	100	78	66	**72**
2	19	500	100	70	72	**71**
3	19	500	100	70	60	**65**
4	19	500	100	80	62	**71**
5	19	500	100	76	64	**70**
6	19	500	100	70	74	**72**
OVERALL AVERAGES OF SIX SUBSETS				**74**	**66.33**	**70.17**

Table 2.13. Results of k-Nearest Neighbors experiments on Australian Dataset

Fold (subset) number	Classification Threshold	# of training data	# of test data	Good Borrower Accuracy (%)	Bad Borrower Accuracy (%)	Average Total Accuracy (%)
1	19	500	100	86	70	**78**
2	19	500	100	84	74	**79**
3	19	500	100	86	82	**84**
4	19	500	100	76	74	**75**
5	19	500	100	88	68	**78**
6	19	500	100	76	62	**69**
OVERALL AVERAGES OF SIX SUBSETS				**82.67**	**71.67**	**77.17**

In Table 2.12 and Table 2.13, classification performances of each subset of the 6-fold cross validated data are shown. ‘Average good borrower accuracy’, ‘average bad borrower

accuracy' and 'average total accuracy' are found by the aritmetical averages of corresponding accuracy values of the 6 subsets presented in the tables. According to k-Nearest Neighbors procedure results, 82.67 per cent of average good borrower accuracy, 71.67 per cent of average bad borrower accuracy and 77.17 per cent of average total accuracy is obtained along with Australian Dataset whereas 74 per cent of average good borrower accuracy, 66.33 per cent of average bad borrower accuracy and 70.17 per cent of average total accuracy is obtained for German Dataset.

2.5. Classification Performances and Experiments

This study aims to investigate credit customer classification ability of Hidden Markov Model. For better understanding and better evaluation of the performance of HMM, the results should have to be compared in a benchmarking manner with other classification. For this reason, k Nearest Neighbors algorithm and Logistic Regression method are also employed for the same purpose. Australian and German datasets are two datasets used in the experiments [21, 22]. For the sake of being identical, exactly the same instances of datasets are utilized with 6-fold cross validation for all three classification methods with same number of training and test samples.

As indicated in previous sections, the experiments are performed with 500 training data for both datasets where 250 of them belong to good borrower class and other 250 of them belong to bad borrower class. Likewise, in all experiments in both datasets, 50 test data belonging to good borrower class and 50 test data belonging to bad borrower class are used as test set, yielding a total of 100.

The performance metrics of the three classifiers can be determined with two kind of criteria. These are classification accuracy, error cost and receiver operating characteristics (ROC) which will be scrutinized in the following sections. The results are also shown in comparative manner in terms of these two metrics.

2.5.1. Classification Accuracy

Classification accuracy, or only accuracy, is a metric about how well a classifier manages to classify the data as correctly as possible. In other words, it is a measure of how correctly or how well the classifier performs.

For better understanding of classification accuracy, it is very useful to look at the contingency table for a binary classification presented in Table 2.14. In the table, 'actual condition' of the test sample represents the real class of the sample whereas 'predicted outcome' refers to the predicted class by the classifier. According to Table 2.14 binary prediction results are named as 'true' and 'false'. 'True' refers to correctly classified entries whereas 'false' refers to incorrectly classified entries. On the other hand, if we extend this situation to credit customer classification case, 'positive' refers to being solvent and 'negative' refers to being insolvent. In this regard,

- The term 'true positive' (TP) means that an actual good borrower is predicted correctly as solvent. In other words, TP corresponds to granting credit to an actual good borrower.

- The term 'false positive' (FP) means that an actual bad borrower is predicted incorrectly as solvent. In other words, FP corresponds to granting credit to an actual bad borrower.

- The term 'true negative' (TN) means that an actual bad borrower is predicted correctly as insolvent. In other words, TN corresponds to rejecting an actual bad borrower.

- The term 'false negative' (FN) means that an actual good borrower is predicted incorrectly as insolvent. In other words, FN corresponds to rejecting an actual good borrower.

Table 2.14. Contingency Table for Binary Classification

		Actual Condition	
		True	***False***
Predicted Outcome	***Positive***	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

In the light of the definitions above, the accuracy is calculated as in equation (2.10) below:

$$\text{Accuracy} = \frac{(\#\text{ of true positives}) + (\#\text{ of true negatives})}{\#\text{ of}(\text{true positives} + \text{false positives} + \text{false negatives} + \text{true negatives})} \quad (2.10)$$

Equation (2.10) says that accuracy is the ratio of true results to all number of the test samples [34]. In other words, accuracy is the number of correct classifications over total number of classifications where accuracy of the 6-fold cross validations is computed by dividing the sum of accuracies of each partition (subset) by six.

For convenience, 6-fold cross validation process involves partitioning the dataset to six subsets such that one of the subsets is kept as the validation data for testing the model and other remaining five subsets become training data. This validation process is repeated six times with each of the six subsets is used exactly once as test data.

Table 2.15 and Table 2.16 show the 6-fold cross validated classification accuracy results of Hidden Markov Model, Logistic Regression and k-Nearest Neighbors for German and Australian datasets respectively.

As seen from Table 2.15, on German Dataset, HMM is the best classifier amongst all with an overall classification accuracy of 71.67 per cent. k-Nearest Neighbors algorithm is the second with a classification accuracy of 70.17 per cent whereas Logistic Regression is the third with an overall classification accuracy of 68.83 per cent.

Table 2.15. Classification Accuracy of HMM, LR and k-NN for German Dataset

German Dataset			
	HMM	LR	k-NN
Overall Acuracy (%)	71.67	68.83	70.17
FP Rate (%)	27.67	16.33	33.67
FN Rate (%)	29	46	26
TP Rate (%)	71	54	74
TN Rate (%)	72.33	83.67	66.33

Table 2.16. Classification Accuracy of HMM, LR and k-NN for Australian Dataset

Australian Dataset			
	HMM	LR	k-NN
Overall Acuracy (%)	84.83	85.0	77.17
FP Rate (%)	16.33	16.33	28.33
FN Rate (%)	14	13.67	17.33
TP Rate (%)	86	86.33	82.67
TN Rate (%)	83.67	83.67	71.67

Table 2.16 shows that for Australian Dataset, Logistic Regression performance is slightly better than HMM classification performance where k Nearest Neighbors is the worst amongst all. Logistic Regression has an overall classification accuracy of 85 per cent where HMM has 84.83 per cent of classification accuracy. On the other hand, k-Nearest Neigbors has an overall accuracy of 77.17 per cent.

As seen from Table 2.15 and Table 2.16, there are also some data belonging to FP, TP, TN and FN rates. According to type of the application some of these parameters should also be as important as overall accuracy. For example, in retail markets where there are so many customers using small amount of loans, rejecting an actual good borrower (FN case) is genarally assumed to be as serious as granting credit to actual bad borrower (FP case). However in commercial credit applications where the amount of the loan is rather high, FP cases are undoubtedly more serious than FN cases [35].

Therefore, HMM is supposed to have sufficient accuracy for credit customer classification problem when compared to conventional methods k-Nearest Neighbors and Logistic Regression.

2.5.2. Classification Error Cost

The goal of classification algorithms is generally accepted as minimizing the expected number of erroneous classifications. It is conventionally assumed that the costs of such various misclassification errors (i.e. FP, FN) are identical. However, in real life applications usually, this is not true. For example, in medical diagnosis, classifying an actual diseased person as healthy is much more serious and costly problem than classifying a healthy person as diseased. Likewise, in commercial credit applications, granting credit to an actual bad borrower is much more vulnerable than rejecting an actual good borrower. Therefore sometimes classification problems with uneven misclassification costs should be handled with more attention. In other words, the act of cost sensitive classification aims to minimize the classification error costs rather than maximizing overall classification accuracy [35].

When focused to credit customer classification problem, there are two types of misclassification errors. First one, false negative (FN) represents the case when an actual good borrower is rejected. It is also called Type I error in statistical process [36]. Second type of error, false positive (FP) represents the case when an actual bad borrower is classified as creditworthy and granted credit. It is also called Type II error in statistical process [36].

In retail credit applications (i.e. credit card applications, auto loans, etc…) it is assumed that the cost of false positives (FP) is taken equivalent to the cost of false negatives (FN) because the number of people applying for credit is rather high and the amount of the loan is not much high. So banks and credit card companies can be more eager to grant credit to retail credit applicants with facing the risk of potential bankruptcies of their customers. That is, retail credit lenders may prefer a classifier which has high true positive (TP) rates since retail credit lenders want to have as many good customers as

possible. Generally, the cost of false positives is counted equivalent to the cost of false negatives for retail market. That is shown as C_{FP}=1 and C_{FN}=1 (unit) keeping in mind that both of the datasets in this study are of retail type. The costs resulting from true positives and true negatives are considered as zero. That is shown as C_{TP}=0 and C_{TN}=0 [35].

In commercial applications, however, the costs of different kinds of errors are significantly different. The banks are more conservative when granting credit in commercial applications where the amount of the loan is rather high. In this case, the cost of false positives is much more than the cost of false negatives. That is banks should avoid granting credit to many customers even at the cost of losing some good borrowers. Therefore banks should prefer using a classifier which gives less false positive rates. Generally, the cost of false positives is counted as five times that of false negatives for commercial applications. That is shown as C_{FP}=5 and C_{FN}=1 (unit). The costs resulting from true positives and true negatives are considered as zero. That is shown as C_{TP}=0 and C_{TN}=0 [35].

Table 2.17 is more illustrative about understanding error types and error costs associated with them. In Table 2.17 the cost are unitless.

Table 2.17. Cost Matrix of Two Class Classification for Credit Applications

	Actual Positive	Actual Negative
Predicted As Positive	True Positive (C_{TP}=0)	False Positive (C_{FP}=1 for retail applications C_{FP}=5 for commercial applications)
Predicted As Negative	False Negative (C_{FN}=1)	True Negative (C_{TN}=0)

The error costs associated with the three classifiers (HMM, LR and k-NN) studied in this thesis are calculated with the formula shown in equation (2.11).

$$\text{Average Error Cost} = \frac{\left[(\#\text{ of FPs}) \times C_{FP} + (\#\text{ of FNs}) \times C_{FN}\right]}{\text{total number of test samples}} \qquad (2.11)$$

The costs due to true negatives and true positives are ignored because C_{TP} and C_{TN} are accepted as zero. The error costs associated with the three classifiers are shown in Table 2.18 below:

Table 2.18 Error Costs

	Australian Dataset			German Dataset		
	HMM	LR	k-NN	HMM	LR	k-NN
Retail Error Cost	0.15165	0.15	0.2283	0.28335	0.31165	0.29835
Commercial Error Cost	0.47825	0.4766	0.7949	0.83675	0.55325	0.97175

As seen from Table 2.18, on German Dataset, Logistic Regression has the lowest error cost with 0.15. Logistic Regression is followed very closely by HMM with an error cost of 0.15165. k-Nearest Neighbors algorithm follows HMM and Logistic Regression with an error cost of 0.2283. On the other hand, for Australian Dataset, HMM has the lowest error cost with 0.28335. HMM is followed by k-Nearest Neighbors with an error cost of 0.29835. Logistic Regression has the highest error cost with 0.31165.

One has to keep in mind that both of the datasets are of retail type. Nonetheless, we have calculated the error costs if the datasets were of commercial type. According to these calculations, Logistic Regression has the lowest error costs whereas k-Nearest Neighbors has the highest error costs. Commercial error costs cannot be used as a valid metric for performance evaluation purposes with given retail datasets; they are just computed for illustrative purposes.

In this study, training data and test data were equally distributed between two classes. That is, we have employed equal number of test and training data from both classes because we wanted all the conditions to be same for both classes. However if someone wants more accuracy for a specific class, then the solution is to use more samples of that class. In other words, one must oversample the cases from that specific class to improve the sensitivity on that class. This is called ‘stratified sampling’ [37]. For example, for commercial credit market, if a lender wants to be more sensitive in false positives, and

wants minimum FP rate, then the lender should use a classifier whose training set contains more samples from insolvent borrower class so that the employed classifier will be more accurate in classifying insolvent credit applicants.

2.5.3. Receiver Operating Characteristics (ROC)

Receiver Operating Characteristics (ROC) is the graphical interpretation of TP rate (sensitivity) vs FP rate (1-specifity) in a binary classification procedure. ROC is considered as one of the optimality criteria in machine learning and data mining. In the ROC space shown in Figure 2.8 below, the optimum point is (0,1) which is also called the *perfect classification point* indicating a TP value equal to one and a FP value equal to zero. The optimality of the classifier is measured according to the ROC distance (d_{ROC}) of the classifier with respect to perfect classification point [39].

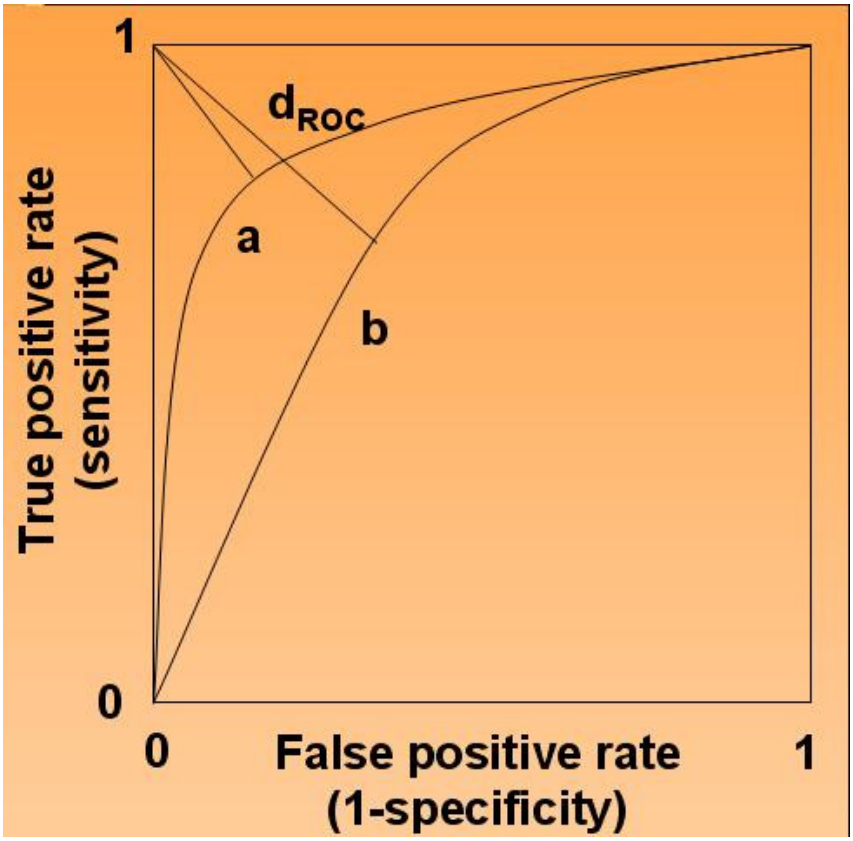


Figure 2.8. ROC Space

ROC distance (d_{ROC}) is calculated with the following equation (2.12).

$$d_{ROC} = \sqrt{(1-TP)^2 + (FP)^2} \tag{2.12}$$

The challenge is to compare the classifiers such that the best is the one with minimum ROC distance. For determining the optimality according to this, the ROC distances of the three classifiers are shown in Table 2.19 for German and Australian datasets. For convenience, the FP and TP rates of the classifiers are also shown in Table 2.15 and in Table 2.16 presented in section 2.5.1.

Table 2.19. ROC Distances

	Australian Dataset			German Dataset		
	HMM	LR	k-NN	HMM	LR	k-NN
d_{ROC}	0.2151	0.2130	0.3321	0.4008	0.4881	0.4254

According to Table 2.19, minimum ROC distance is possible for LR which is followed by HMM on Australian dataset. This means LR is the most optimal classifier among all whereas k-NN is the worst. On the other hand on German dataset, HMM is the most optimal classifier. It is followed by LR which is followed by k-NN.

2.6. Triage Strategy

Sometimes, it is useful to have a third option on binary classification. In addition to two classes in a binary classification, a third option can be ‘undecided’ choice. For some of the test samples, it is not very convenient to distinguish the class especially when the sample point stands very close to decision boundary of the classifier. For such a case, the test sample is labeled as ‘undecided’ meaning that the classifier cannot classify this test sample. In other words ‘undecided’ option somehow corresponds to lack of information about the class of the test sample. In this case, generally a human expert is to investigate the case and make a decision about the test sample himself [37]. For example in the case of a credit approval operation, whether a classifier generates a ‘undecided’ option for a credit

applicant, then an expert credit officer should have to evaluate the applicant with a closer scrutiny by himself.

Especially in the case of noise existence in a dataset, the samples around the decision boundary become very risky for classification. In other words, samples around the decision boundary are more likely to be misclassified. In literature the utilization of undecided option in classifiers is employed in credit risk analysis and image processing applications [37], [39-40].

In order to determine which data can have the option 'undecided', there must be some confidence threshold around the decision boundary. If test sample is beyond the confidence threshold then it can be classified as 'undecided'. On the other hand, samples under the confidence threshold can be confidently identified for binary classification. The confidence threshold can be determined by trial and error as well as looking into classification database and acquired experience. Taking a broader boundary region (no decision region) should lead to less number of misclassification errors because broader boundary region encloses more test samples in no decision region. This causes less number of samples to be strictly classified with more test samples left unclassified. Taking a narrower confidence threshold undoubtedly leads to more misclassification errors whereas more number of test samples can be incorporated into classification process [40].

In this study, triage strategy is applied to HMM and LR processes. It is seen that triage strategy really eliminated risky data for classification and gave more aacurate classification performances. However as expected, test samples around the decision boundary are left unclassified and this led to the fact that less number of test samples could be tested with the classifier and more samples are left to human experts for classification.

2.6.1. Triage Strategy in HMM

According to our poposed HMM classification procedure, there are two models. First model, named as good model is trained with the training samples belonging to good borrowers. Second model, namely bad model, is trained with training samples of bad

borrowers. Once both the models are trained, log-likelihood values associated with both models (good LL and bad LL) are computed for every test sample. Classification is done according to the ratio of these two log-likelihood values as explained in section 2.2.3. Keeping in mind that log-likelihood values are negative, the decision boundary is the contition where good LL and bad LL are equal to each other.

The idea of applying triage strategy for HMM is not very complicated. Since the decision boundary is the case when good LL is equal to bad LL, the decision boundary can also be expressed as the ratio of good LL to bad LL. The value of this log-likelihood ratio (LL ratio), also shown in equation 2.13 below, is obviously 1.0. Confidence region is set around this LL ratio (log likelihood ratio). Since log-likelihood values of the test samples associated with the trained models are negative, LL ratio will be greater than one for the customers predicted as bad and will be smaller than one for the customers predicted as good.

$$\text{LL ratio} = \frac{\text{good LL (log - likelihood associated with good model)}}{\text{bad LL (log - likelihood associated with bad model)}} \qquad (2.13)$$

At first, the confidence threshold is selected as 0.01 meaning that the test samples having LL ratios between 1+0.01 and 1-0.01 are not classified and left undecided. Secondly, confidence threshold is selected as 0.015 which means that test samples having LL ratios between 1+0.015 and 1-0.015 are not incorporated into the classification process and labeled as ‘no decision’. Finally, the risky region around the boundary LL ratio=1 ,is more broadened with a threshold of 0.05. The results are presented in Table 2.20 for German dataset and in Table 2.21 for Australian dataset.

The interpretation of Table 2.20 and Table 2.21 is, in fact, straightforward. For example, in Table 2.20, the overall average accuracy of all 6 subsets of the cross validated German data is given as 71.67 per cent as also declared in Table 2.4 in section 2.2.3. This is the result before triage strategy is applied to the system. When the triage strategy is applied with a threshold of 0.01, an average of 4.9 per cent of the test samples are left undecided. More precisely, the classifier did not classify 4.9 per cent of the test data, since that undecided data stay very close to boundary region.

Table 2.20. Results of Triage Strategy of HMM on German Dataset

Fold (subset) number	Average Total Accuracy (%)	Average Total Triage Accuracy (%) [threshold:0.01]	Average 'Undecided' Data (%)[threshold:0.01]	Average Total Triage Accuracy (%) [threshold:0.015]	Average 'Undecided' Data (%)[threshold:0.01]	Average Total Triage Accuracy (%) [threshold:0.05]	Average 'Undecided' Data (%)[threshold:0.05]
1	**76**	77.27	3.2	76.75	5.4	78.75	20
2	**78**	79.27	6.4	79.87	9.6	80.14	28.6
3	**70**	72.04	7	71.92	11	77.14	32
4	**67**	68.89	3.6	70.14	7	73.57	22.8
5	**65**	65.52	4.4	66.93	9.4	71.7	25.8
6	**74**	74.78	4.8	75.43	7.2	75.9	22
Overall Average of 6 subsets	**71.67**	**72.94**	4.9	**73.51**	8.27	**76.11**	25.2

Table 2.21. Results of Triage Strategy of HMM on Australian Dataset

Fold (subset) number	Average Total Accuracy (%)	Average Total Triage Accuracy (%) [threshold:0.01]	Average 'Undecided' Data (%)[threshold:0.01]	Average Total Triage Accuracy (%) [threshold:0.015]	Average 'Undecided' Data (%)[threshold:0.01]	Average Total Triage Accuracy (%) [threshold:0.05]	Average 'Undecided' Data (%)[threshold:0.05]
1	**83**	83.37	6.2	84.06	7.2	85.45	13.4
2	**85**	87.09	4	87.45	4.4	87.83	8
3	**86**	86.68	2.4	86.75	3.4	88.45	8.4
4	**84**	85.21	9.4	85.91	10.6	86.92	14.4
5	**87**	87.07	1	87.2	1.6	88.07	7.8
6	**84**	86.53	5	86.05	5.4	89.13	10.2
Overall Average of 6 subsets	**84.83**	**86.01**	4.67	**86.25**	5.43	**87.65**	10.37

Therefore, 100% - 4.9% = 95.1% of the test data are exposed to classification procedure. The overall average accuracy of this 95.1 per cent of the data is achieved as 72.94 per cent when the threshold is 0.01. If the threshold is increased to 0.015, the percentage of undecided data is 8.27 per cent which means that 91.73 per cent of the data are exposed to classification procedure. The accuracy is 73.51 per cent when the threshold is 0.015. Finally when the threshold is increased to 0.05 the percentage of undecided data is reached to 25.2 per cent with an overall average accuracy of 76.11 per cent.

Similar results are also seen in Table 2.21 in the sense that if boundary region is widened, more test samples are enclosed in this region and they become out of classification. In other words, increasing boundary region will remove more number of risky data. This leads to the fact that classification is employed with less risky remaining data. This means that broader boundary region causes more accurate results with more data left undecided.

2.6.2. Triage Strategy in Logistic Regression

Logistic Regression is a method that takes the total contribution of all input features as input. The output is a probability of an occurrence of an event. The input features are somehow predictive variables and they are transformed according to equation (2.6). The output of Logistic Regression process is accepted as a probability and has a value between 0 and 1.

The classification is employed with a decision boundary whose value is 0.3775 ($\approx 1-\frac{1}{1+e^{-z}}$ where z=0.5) as explained in section 2.3.2. At first the confidence threshold is selected as 0.3 meaning that the decision bondary consists of the region between $1-\frac{1}{1+e^{-(0.5+0.3)}}$ ($\approx$0.31) and $1-\frac{1}{1+e^{-(0.5-0.3)}}$ ($\approx$0.45). In this case, test samples having Logistic Regression outputs between 0.31 and 0.45 are not employed in classification process and they are labeled as ‘undecided’. The samples having values greater than 0.45 are classified as insolvent borrowers and samples having values smaller than 0.31 are classified as solvent borrowers. Secondly, confidence threshold is selected as 0.5. Finally,

the risky region around the boundary is more broadened with a threshold of 0.7. The results are presented in Table 2.22 for German dataset and in Table 2.23 for Australian dataset.

Table 2.22. Results of Triage Strategy of Logistic Regression on German Dataset

Fold (subset) number	**Total Accuracy (%)**	**Total Triage Accuracy (%)** [threshold:0.3]	**'Unecided' Data (%)**[threshold:0.3]	**Total Triage Accuracy (%)** [threshold:0. 5]	**'Undecided' Data (%)**[threshold:0.5]	**Total Triage Accuracy (%)** [threshold:0.7]	**'Undecided' Data (%)**[threshold:0.7]
1	**75**	77.27	12	80.52	23	81.16	31
2	**69**	72.94	15	78.26	31	78.46	35
3	**59**	62.64	9	66.67	28	72.13	39
4	**71**	73.56	13	78.48	24	73.53	32
5	**70**	75.58	14	70.67	21	78.87	29
6	**66**	67.86	16	75	25	72.72	34
Overall Average of 6 subsets	**68.33**	**71.59**	13.17	**75**	25.33	**76.25**	33.33

The interpretation of Table 2.22 and Table 2.23 is, in fact, straightforward. For example, in Table 2.22, the overall average accuracy of all 6 subsets of the cross validated Australian data is given as 68.33 per cent as also declared in Table 2.10 in section 2.3.2. This is the result before triage strategy is applied to the system. When the triage strategy is applied with a threshold of 0.3, an average of 13.17 per cent test sample is left undecided. More precisely, the classifier did not classify 13.17 per cent of the test data, since that undecided data stay very close to boundary region.

Therefore, 100% - 13.17% = 86.83% of the test data are exposed to classification procedure. The overall average accuracy of this 86.83 per cent of the data are achieved as 71.59 per cent when the threshold is 0.3. If the threshold is increased to 0. 5, the percentage of undecided data is 25.33 per cent which means that 74.67 per cent of the data are exposed to classification procedure. The accuracy have come out to be 75 per cent when

the threshold is 0.5. When the threshold is increased to 0.7 the percentage of undecided data is reached to 33.33 per cent with an overall average accuracy of 76.25 per cent.

Table 2.23. Results of Triage Strategy of Logistic Regression on Australian Dataset

Fold (subset) number	**Total Accuracy (%)**	**Total Triage Accuracy (%)** [threshold:0.3]	**'Undecided' Data (%)**[threshold:0.3]	**Total Triage Accuracy (%)** [threshold:0. 5]	**'Undecided' Data (%)**[threshold:0.5]	**Total Triage Accuracy (%)** [threshold:0.7]	**'Undecided' Data (%)**[threshold:0.7]
1	**81**	84.95	7	87.06	15	86.9	16
2	**80**	81.72	7	82.42	9	82.42	9
3	**88**	88.54	4	90.11	9	91.86	14
4	**87**	90.22	8	90.91	12	92.77	17
5	**90**	90.53	5	91.30	8	91.86	14
6	**84**	88.42	5	89.25	7	89.01	9
Overall Average of 6 subsets	**85**	**87.41**	6	**88.52**	10	**89.06**	13.17

Similar results are also seen in Table 2.23 in the sense that if boundary region is widened, more test samples are enclosed in this region and they become out of classification. In other words, increasing boundary region will remove more number of risky data. This leads to the fact that classification is employed with less risky remaining data. This means that broader boundary region causes more accurate results.

3. PROBABILITY OF DEFAULT MODELING PROCEDURE

Credit risk analysis was, at first, consisting of binary classification of credit customers. The purpose of such a classification is to distinguish between potential solvent and insolvent borrowers at credit application stage prior to granting credit. According to the classification result, a credit applicant is granted credit or not. However this kind of classification is supposed to be insufficient for modern credit risk analysis. For more elaborate and robust risk evaluation, the importance of risk-rating of credit applicants by credit lenders is emphasized in many academic and financial publications [2]. For example, consultative Basel Accords strongly advises the risk-rating of even small firms [15].

Probability of Default (PD) Modeling is known to be one of the most popular risk computation methods. In fact probability of default value is simply a probability that a credit applicant should have a bankruptcy and should not manage to pay his/her debts in a timely manner [17]. Therefore, basically customers having higher PD values are more likely to have a bankruptcy. This situation discourages banks and other credit lenders to grant credits to such borrowers. More precisely, banks should avoid granting credit to high risky borrowers or may prefer to grant less amount of credit to them. On the other hand, customers having less PD measures are considered as more reliable by the banks. It means that it is safer to grant credit to such customers. Hence PD is a measure indicating the creditworthiness of a credit borrower. In other point of view, PD value is a measure of how strong a customer belongs to solvent or insolvent class with his financial potential.

Probability of default is important in not only credit granting decision to borrowers but also in adjustment of capital holding due to a granted credit. If a lender assigns a higher PD value for a credit packet, then more capital must be held for safety. However this leads to redundant capital holding which can be considered as costly. Conversely, if a lender assigns a lower PD value for a credit packet then, the lender is actually facing an unplanned risk which, in turn, may get the bank into financial troubles. Therefore, accurate PD modeling is very crucial in the banking aspect [17].

PD modeling concept works in the sense that the model accepts the attributes of credit applicants as inputs and converts these inputs to a number between 0 and 1 corresponding to a probability that a bankruptcy should occur and the customer should not pay his/her debts in a timely manner. Figure 3.1 below explains the concept of PD modeling.

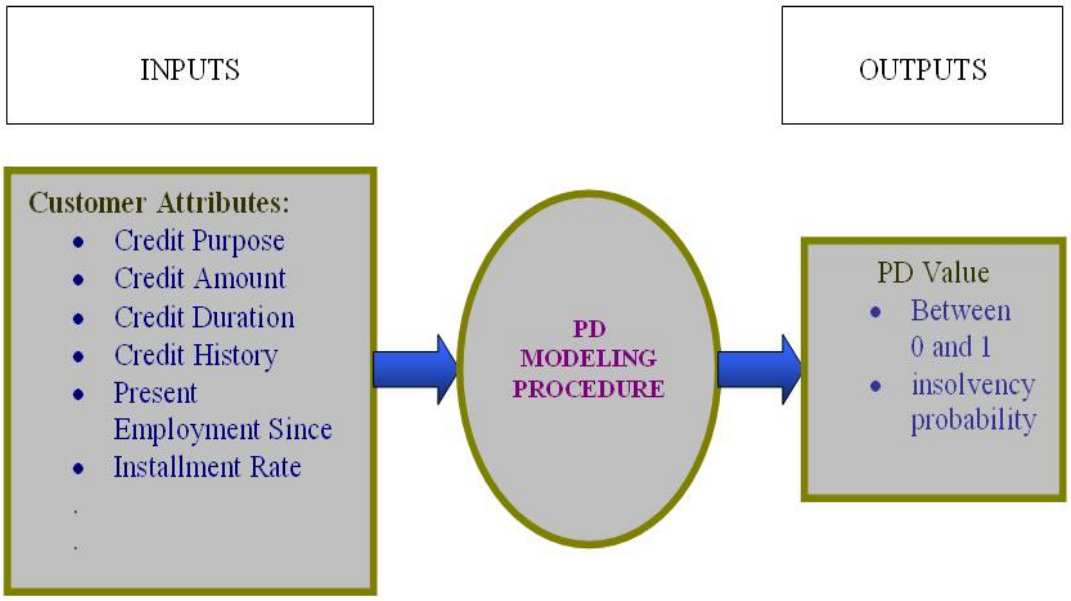


Figure 3.1. The concept of PD modeling

In the following two sections PD modeling ability of Logistic Regression and HMM is explained. Performances of both models are compared with each other to see whether newly proposed PD modeling procedure of HMM is sufficiently good for credit risk analysis.

3.1. PD Modeling with Logistic Regression

Logistic Regression model is used for predicting the probability of occurrence of an event and generally used in statistics. The inputs are considered as the predictive variables influencing the probability of a certain event which is the output of Logistic Regression [30]. Therefore, Logistic Regression itself directly generates probability results. Therefore

it is very straightforward to find PD values with Logistic Regression because what Logistic Regression gives is purely the default bankruptcy probabilities associated with credit borrowers and their financial predictive information.

In section 2.3 Logistic Regression is scrutinized in details for credit classification applications. Total contribution of all input features, denoted as z, is incorporated in the transformation given in equation (3.1) below: as well as in equation (2.6):

$$p = f(z) = \frac{1}{1+e^{-z}} \tag{3.1}$$

In equation (3.1), p, is the output of Logistic Regression operation f(z). p is, as stated, the probability of insolvency. More precisely, p directly holds for PD value. Therefore, Logistic Regression has the advantage that it can directly give PD value. The transformation function in equation (3.1), f(z) has a wide input range between minus infinity and plus infinity. On the other hand, f(z), has outputs in the range (0,1) so that the transformation corresponds to a probability value concerning the predictive input variables given in equation (2.7). The challenge is to fit the data into the logistic curve shown in Figure 2.5 with the given transformation by using the given training set.

The parameters optimized in curve fitting are the regression coefficients associated with each input feature. The transformation function in equation (2.6) can also be expressed as in equation (3.2) below:

$$\ln\left(\frac{p}{1-p}\right) = z = \beta_0 + \beta_1\chi_1 + \beta_2\chi_2 + \ldots + \beta_k\chi_k \tag{3.2}$$

Equation (3.2) can be interpreted as the process of linear regression of input features to obtain natural logarithms of odds ratio concerning the bankruptcy of a customer. The input z consists of total contribution of all input feature vectors. In our case, input features are the attributes shown in Table 2.1 and Table 2.2.

In classification procedure we have employed a probability threshold for classifying the test samples. Test data having probabilities greater than the classification threshold are considered as probable insolvent borrowers whereas the data having probabilities less than the threshold are considered as probable solvent borrowers. The probability output of Logistic Regression, *f(z)*, is directly counted as the probability of default (PD) value of the relevant customer. Therefore, the probability values found for classification in section 2.3 are nothing but PD values.

PD computations with Logistic Regression are handled for both datasets, each having a training set of 500 samples and a test set of 100 samples. PD values are computed based on 6-fold cross validation principle. Detailed results are given in section 3.3 below.

3.2. PD Modeling with HMM

One of the major aims of this thesis study is to build an appropriate and robust procedure for PD modeling. In classification case two models were formed. First model, namely good model, is trained with the data belonging to good borrowers. Second model, namely bad model, is trained with the data belonging to bad borrowers. Once these two models are trained, next step is to compute the log-likelihood values of the test sample associated with both models. The class of the test sample is determined according to the log-likelihood (LL) values. If LL value associated with good model is greater than the LL value associated with bad model then the sample is classified as good borrower. Conversely if LL value associated with bad model is greater than the LL value associated with good model then the sample is classified as bad borrower. The classification procedure with HMM is explained with full details in section 2.2.3.

Up to this point, we have got two trained models, and the log-likelihood values of the test samples associated with these two models. Therefore one has to employ some other parameters to jump to probabilities concerning insolvency. Logistic Regression is very easy going method in this sense because it directly gives the PD measures. However HMM does not have direct outputs that can be used as PD values. Therefore one has to employ a function to transform HMM outputs into PD values.

As known from section 2.2.3 for every test sample, log-likelihood (LL) values associated with the good model and the bad model are computed. The ratio of LL value associated with the good model (good LL) to LL value associated with the bad model (bad LL) can be considered as an appropriate parameter for PD modeling. We call this ratio as LL ratio in short form. LL ratio is the first parameter that we use in transformation function and it is also demonstrated in equation (2.13) in Section 2.

If a test sample is predicted as good, then it means that good LL value is bigger than bad LL value for the test sample. Since LL values are negative for HMMs, a test sample predicted as good has a LL ratio smaller than one. If LL ratio is equal to one, in this case the test sample is right on the boundary between good model and bad model. For such a case, this test sample is supposed to have a PD value equal to 0.5 because if a borrower is considered as a fully good customer then the risk is zero and this customer is expected to have a PD value of 0. On the other hand, if a borrower is considered as a fully bad customer then the risk is one and this customer is expected to have a PD value of one. Hence it is assumed that a LL ratio bigger than one corresponds to a customer whose PD value is more than 0.5.

Therefore, the assumptions are as follows:

- LL ratio = 1 => PD value = 0.5
- LL ratio < 1 => PD value < 0.5
- LL ratio >1 => PD value > 0.5
- If LL ratio increases, PD value increases too.

In the light of these assumptions, a linear regression is to be employed for converting the LL ratios to PD values. However, one more parameter is needed for linear regression. This second parameter is determined as empirical PD (ePD) value which is an immature form of PD values [41]. Therefore linear regression will bring out PD values as estimations of ePD measures under least squares method. Step by step procedure of computation of ePD value, which can be considered as a pre-probability of insolvency, is given as follows:

- Compute LL ratios for each training data just like in the case of test data.
- For the i^{th} test sample, find the training sample having the closest LL ratio to that of i^{th} test sample and call this training sample as target training sample.
- Select h number of nearest neighboring training samples having LL ratios greater than that of target training sample, and select another h number of nearest neighboring training samples having LL ratios smaller than the LL ratio of target training sample. Including the target training sample also, will make a total of (2h+1) entries of nearest training samples associated with the i^{th} test sample.
- Last step of computing ePD value associated with i^{th} test sample is expressed in equation (3.3) as below.

$$ePD_i = \frac{\#\text{ of insolvencies among } (2\text{h}+1) \text{ nearest training neighbors}}{2\text{h}+1} \tag{3.3}$$

As seen from equation (3.3), empirical PD for the i^{th} test sample is found by the ratio of the total insolvencies around the target training sample.

Once ePD values associated with each test sample is found along with corresponding LL ratios, now it comes to employing first order least squares linear regression to estimate PD values. When the linear regression parameters are computed, each PD value associated with corresponding LL ratio is found for every test data. This last process will bring us the final PD values for each test sample.

One bottleneck of PD modeling with HMM is that the computational cost can be considered as high because for every test sample, LL ratios associated with the entire training set are computed and sorted in order to find the nearest ones. In PD computations with HMM, h is taken as 15, therefore a total of 31 (2xh+1) nearest training samples is used for ePD calculations. It is also important to notice that the term ‘nearest’ refers to having nearest LL ratios in our case.

3.3. Alternative Way of PD Modeling by HMM

As a second way, we propose another method for PD modeling procedure. According to this second model, we utilize the posterior probabilities in order to achieve a reasonable default probability. As explained in previous section we have two models: good model (model G) and bad model (model B). A credit applicant, meaning a test sample, belongs to either one of these two models. Using Bayes' conditional probability relation, we can write the following equation:

$$P(B\backslash X) = \frac{P(B,X)}{P(X)} = \frac{P(X\backslash B)P(B)}{P(X\backslash G)P(G) + P(X\backslash B)P(B)} \tag{3.4}$$

In this equation, X is the observation sequence of the credit applicant associated with the test sample. $P(B\backslash X)$ is nothing but the desired PD value because it represents the probability that the bad models is valid given that the observation sequence have come out to be X. More precisely, this is the probability of X's belonging to an actual bad borrower. On the other hand, $P(X\backslash B)$ is the probability of occurrence of X given the bad model. Likewise P(X\G) is the probability of X given the good model. Finally, $P(B)$ is the probability that the valid model is bad model and $P(B)$ is taken equal to 0.5 because we have employed equal number of good and bad training samples. Due to the same reason, $P(G)$, the probability that the valid model is good model, is also taken equal to 0.5. Therefore the above equation can be reduced as below:

$$PD = P(B\backslash X) = \frac{P(X\backslash B)}{P(X\backslash G) + P(X\backslash B)} \tag{3.5}$$

Equation (3.5) is more practical and easier method for PD modeling procedure. Its computational cost is rather low when compared to first proposed method in section 3.2. It does not require such a sorting algorithm; it does not require any further computations regarding the training data. More than that, a linear regression process is not needed to convert the ouputs to default probability values. The output of this new algorithm directly

gives PD values. And finally this method gives better PD results as shown in the next section.

3.4. PD Modeling Performances

PD modeling experiments are employed with 250 good training data and 250 bad training data just like in classification experiments with 6-fold cross validation. Test data consist of 50 samples of good borrowers and 50 samples of bad borrowers. In Table 3.1 and Table 3.2 PD values associated with both datasets are given for HMM and Logistic Regression.

Table 3.1. PD Modeling Results with HMM

Subset	German Dataset				Australian Dataset			
	Good borrower average PD		**Bad borrower average PD**		**Good borrower average PD**		**Bad borrower average PD**	
	1st Method	**2nd Method**	**1st Method**	**2nd Method**	**1st Method**	**2nd Method**	**1st Method**	**2nd Method**
1	0.3493	0.3046	0.6389	0.6521	0.2821	0.1852	0.7228	0.7455
2	0.4251	0.3794	0.6446	0.7056	0.2565	0.1915	0.7732	0.8007
3	0.4341	0.3552	0.6411	0.6326	0.2452	0.1694	0.7822	0.8169
4	0.4261	0.3586	0.6124	0.6487	0.3177	0.2167	0.7732	0.8140
5	0.4453	0.4445	0.6323	0.6699	0.2416	0.1991	0.7623	0.8227
6	0.355	0.2975	0.5701	0.6113	0.2203	0.2159	0.7743	0.8123
Average of All	**0.4058**	**0.3566**	**0.6233**	**0.6534**	**0.2605**	**0.1963**	**0.7647**	**0.802**

As seen in Table 3.1 and in Table 3.2, all the computed PD values are shown for each subset of 6-fold cross validated data. A good PD model should have to assign higher PD values to probable bad borrowers and should assign lower PD values to probable good borrowers. However, it is not possible to have a consistent performance comparison due to limitations of the datasets. Instead, the following observations are worthy of noticing.

Table 3.2. PD Modeling Results with Logistic Regression

Subset	German Dataset		Australian Dataset	
	Good borrower average PD	**Bad borrower average PD**	**Good borrower average PD**	**Bad borrower average PD**
1	0.3490	0.6512	0.1950	0.7369
2	0.3927	0.6374	0.0858	0.5470
3	0.4138	0.6219	0.1889	0.7909
4	0.3685	0.6069	0.2095	0.8174
5	0.3343	0.5902	0.1866	0.7943
6	0.4221	0.6308	0.2282	0.8040
Average of All	**0.3801**	**0.6231**	**0.1823**	**0.7484**

In Australian dataset, Logistic Regression gives an average PD value of 0.1823 whereas first HMM method gives an average PD value of 0.2605 and second HMM method gives an average PD value of 0.1963 for solvent borrowers. Therefore Logistic Regression performs better than both HMM methods for good borrowers in Australian dataset. On the other hand, Logistic Regression gives an average PD value of 0.7484 for bad borrowers whereas first HMM method gives an average value of 0.7647 and second HMM method gives an average value of 0.802. Hence, this time, HMM performs better for bad borrowers.

In German dataset, Logistic Regression gives an average PD value of 0.3801 whereas first HMM gives an average PD value of 0.4058 and second HMM method gives an average PD value of 0.3566 for solvent borrowers. Therefore Logistic Regression performs a little bit better than first HMM method but a little bit worse than second HMM method for good borrowers in German dataset. On the other hand, Logistic Regression gives an average PD value of 0.6231 for bad borrowers whereas first HMM method gives an average value of 0.6233 and second HMM method gives an average value of 0.6534. Hence, this time, HMM performs better for bad borrowers.

Therefore, HMM should be an appropriate method for PD modeling purposes as well as Logistic Regression which is a known to be a conventional method. It is also seen that HMM is a consistent measure for credit decision process in the sense that it gives higher default bankruptcy probabilities (simply PD) to actual bad borrowers and assigns low probabilities to actual good borrowers as seen in summary table, Table 3.3 as in Table 3.1 and Table 3.2.

Table 3.3. Summary of PD modeling results for both datasets

German Dataset			
	HMM 1st method	**HMM 2nd method**	**LR**
Average Good Borrower PD Value	0.4058	0.3566	0.3801
Average Bad Borrower PD Value	0.6233	0.6534	0.6231
Australian Dataset			
	HMM 1st method	**HMM 2nd method**	**LR**
Average Good Borrower PD Value	0.2605	0.1963	0.1823
Average Bad Borrower PD Value	0.7647	0.802	0.7484

4. CONCLUSION

In this study, credit risk analyzing ability of Hidden Markov Model is scrutinized in terms of classification of credit customers and probability of default (PD) modeling. Classification of credit applicants and PD modeling are two trustworthy measures of credit risk analysis. For the purpose of comparison of classification performances, Logistic Regression (LR) and k-Nearest Neighbors (k-NN) are two conventional classification methods that are together utilized in this study. On the other hand, PD modeling ability of HMM is compared with that of Logistic Regression.

Credit classification simply distinguishes between solvent and insolvent credit applicants prior to granting credit. On the other hand, instead of binary classification, PD models assign default bankruptcy (insolvency) probabilities between 0.0 and 1.0 to credit borrowers concerning their credit applications.

German credit dataset and Australian dataset are two public access databases used in this study. Both of the datasets are containing data concerning applications of retail type. The limitations due to these two retail datasets might be handled and overcome by using larger datasets of from various credit lenders. And also commercial datasets should also be used for investigating the performance as well as retail datasets. However, since credit datasets contain private information concerning borrowers and the lenders, banks are generally not very eager to share this type of data. For this reason we are said to be confined to aforementioned German and Australian datasets.

Classification experiments are employed with 6 fold cross validation in both datasets using HMM, LR and k-NN. Classification performances are interpreted in terms of classification accuracy and error cost. In German dataset, HMM gives best accuracy followed by k-NN. In Australian dataset, on the other hand, LR has the best accuracy followed by HMM. Keeping in mind that both datasets utilized in the experiments are retail datasets, HMM gives least retail error cost in German dataset. HMM is followed by

LR. On the other hand, LR gives least retail error cost for Australian dataset where LR is followed by HMM.

Instead of strict binary classification, triage strategy is also introduced as a third option. This third option offers a choice of 'undecided' data meaning that there is no sufficient clue for determining the class of the test sample. After removing undecided customers, it is noticed that classification accuracy is improved among the remaining data. In classification of credit customers, if 'undecided' option is assigned to the customer, it means that the decision about the customer is left to human experts. However it is also important not to leave too much customers to human experts.

PD modeling ability of HMM is investigated in the second part of this study. Logistic Regression is compared to HMM in terms of PD modeling performance. However PD modeling with HMM is not very straightforward as in the case of Logistic Regression because Logistic Regression directly gives PD values. On the other hand some extra process must be applied to convert the outputs of HMM to PD values. For this purpose, first order (linear) regression is employed between empirical PD measures and the ratios of log-likelihood values concerning the trained models of solvent and insolvent borrowers.

In PD modeling experiments it is seen that HMM gives higher PD values for bad borrowers in both datasets. On the other hand, Logistic Regression gives lower PD values to solvent borrowers in both datasets. Therefore the results show that HMM can be used for PD modeling as an alternative to conventional PD modeling methods like Logistic Regression. Therefore HMM is said to be more sensitive to more risky borrowers in the sense that it gives higher PD values (bankruptcy probabilities) for insolvent borrowers.

Credit risk assessment is a very crucial area of growing interest because of financial crises of the 1980's and 90's [42]. Moreover the latest global crisis of 2008 broke out due to non returning mortgage loans and corruption of big banks. This financial picture once more underlines the importance and vitality of credit risk assessment. For this reason researchers have been studying this issue for tens of years. In this study we have scrutinized the credit risk scoring ability of HMM in terms of classification and PD modeling. In the future, reduction of computational cost of HMM should be studied for

credit risk analysis. Especially, initial model selection of the HMM process should be improved to converge faster and better for more accurate risk modeling. One more future work should be using HMM together with other classifiers. In this way, more accuracy, more optimality and less cost should be possible for classification process by combining the power of more classifiers.

APPENDIX A: DOCUMENT FOR HMM TOOLBOX OF MATLAB

Matlab's HMM tool developed by Kevin Murphy is one of the most popular HMM tools so far and it is widely utilized in this thesis study. Its usage is considered as rather simple. However due to Matlab's slow operating speed, some long iterations may be cumbersome and time consuming. The toolbox can be downloaded at http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html.

Another famous HMM toolkit is known to be HTK toolkit which is widely used for classification puposes and speech processing by researchers. HMM training testing and complex HMM systems can be built with HTK. HTK consists of a set of library modules and tools available in C source form. In order to use the toolkit, source code must be compiled with an ANSI C compiler [43]. Due to its detailed and difficult syntax it is not preferred in this study.

Usage of MATLAB HMM Toolbox:

First of the the directory of the toolbox should be incorporated in the workspace of MATLAB as follows (assuming that the directory of toolbox is @ C:\HMMall):

```
>> addpath(genpath('C:\HMMall'))
```

In order to initialize a random Hidden Markov Model, one has to use the following expressions:

```
>> prior_good_1 = normalise(rand(Q,1));
>> transmat_good_1 = mk_stochastic(rand(Q,Q));
>> obsmat_good_1 = mk_stochastic(rand(Q,O));
```

In the expressions above, Q is the number of hidden states given by the user for the model and O is the total number of observation symbols in the model. '*normalise*' function determines the initial hidden state probabilities of the system because '*rand(Q,1)*' generates a vector of size Q in which the values of the elements are randomly assigned

between 0 and 1. ‘normalise’ function normalizes the vector ‘*rand(Q,1)*’ so that the initial state probabilities π_i, are found.

The expression, ‘*mk_stochastic(rand(Q,Q))*’, is to generate the transition matrix of the initial model. Since ‘*rand(Q,Q)*’ generates a square matrix of Q by Q with each element between 0 and 1, ‘*mk_stochastic*’ function normalises every row of the matrix ‘*(rand(Q,Q))*’ making it nothing but the state transition matrix (eqn. 2.2 and 2.3) for our initial model. Likewise, the expression, ‘*obsmat_good_1 = mk_stochastic(rand(Q,O))*’, aims to find the observation symbol probability distribution (eqn. 2.4) for each state. In other words, for each state, the observation symbol distributions are computed with this expression.

Once the initial models are generated next step is to train the model such that the model parameters best fit the given data by using Baum Welch algorithm [25-26]. The expression ‘*[LL_good, prior_good_2, transmat_good_2, obsmat_good_2] = dhmm_em(training_data(1:250,:), prior_good_1, transmat_good_1, obsmat_good_1, 'max_iter', iter_count, 'tresh', treshold*)’ computes the final model parameters. Training data, initial model (initial state probabilities, state transitions and observation symbol distributions), number of maximum iterations and the convergence threshold are given to the function ‘*dhmm_em*’ as input arguments. The output of this function is the new model with new initial state probabilities, new state transitions matrices and new observation symbol distributions.

Finally, after the models are trained, one has to find the log-likelihood values of test samples associated with trained models. Log-likelihood values indicate how a test sequence belongs to a certain model. In order to find the log-likelihood values, one has to employ the expression ‘*loglikTest(j,i,1) = dhmm_logprob(test_data(i,:), prior_good_2, transmat_good_2, obsmat_good_2)*;’. This expression computes the log-likelihood value of each test sequence with respect to the finalized trained model. The function ‘*dhmm_logprob*’ uses the test sequences and the relevant model (initial state distribution, state transition matrix and observation symbol distributions) as input arguments and gives log-likelihood values as outputs.

REFERENCES

1. Basel Committee on Banking Supervision, *Principles for the Management of Credit Risk*, http://www.bis.org, 1999.

2. Allen, L., G. DeLong and A. Saunders, "Issues in Credit Risk Modeling of Retail Markets", *Journal of Banking and Finance,* Vol. 28, pp. 727–752, 2004.

3. Wikipedia, *Credit Score,* http://en.wikipedia.org/wiki/Credit_score.

4. Wiginton, J. C., "A Note on the Comparison of Logit and Discriminant Models of Consumer Credit Behavior", *Journal of Financial and Quantitative Analysis,* Vol. 15, No. 3, pp. 757–770, 1980.

5. Kim, Y. H. and V. Srinivasan, "Credit granting: A Comparative Analysis of Classification Procedures", *Journal of Finance,* Vol. 42, No. 3, pp. 665–681, 1987.

6. Chang, K. C., R. Fung, A. Lucas, R. Oliver and N. Shikaloff, "Bayesian Networks Applied to Credit Scoring", *IMA Journal of Mathematics Applied in Business and Industry,* Vol. 11, pp. 1–18, 2000.

7. Baesens, B., R. Castelo, M. Egmont-Petersen and J. Vanthienen, *Learning Bayesian Network Classifiers for Credit Scoring Using Markov Chain Monte Carlo Search*, http://www.cs.uu.nl/research/techreps/repo/CS-2001/2001-58.pdf, 2001.

8. Schebesch, K.B. and R. Stecking, *Support Vector Machines for Credit Scoring: Extension to Non Standard Cases*, Institut für Konjunktur und Strukturforschung, Universität Bremen, Bremen, 2005.

9. West, D., "Neural Network Credit Scoring Models", *Computers & Operations Research,* Vol. 27, pp. 1131–1152, 2000.

10. Hand, D. J. and W. E. Henley, "A k-Nearest-Neighbour Classifier for Assessing Consumer Credit Risk", *The Statistician,* Vol. 45, No. 1, pp. 77–95, 1996.

11. Gürgen, F., M. E. Kaya and N. Okay, "A Comparison of Support Vector Machines and Logistic Regression for Credit Risk Analysis", *Intl. Journal of Information Technology (ITIC)*, 2008.

12. Kaya, M. E., *Credit Risk ModelingUsing Machine Learning Techniques*, M.S. Thesis, Boğaziçi University, 2006.

13. Cai, Y., S. Pan, K. Sun, R. Xu and B. Wu, *Hidden Markov Model for Corporate Default Risk Analysis*, https://ccnet.stanford.edu/cgi-bin/course.cgi?cc=msande347&action=handout_download&handout_id=ID121349839313972, Stanford University, 2008.

14. Banachewicz, K. and A. Lucas, "Quantile Forecasting for Credit Risk Management Using Probably Misspecified Hidden Markov Models", *Journal of Forecasting,* Vol. 27, pp. 566–586, 2008.

15. Basel Committee on Banking Supervision, *The New Basel Capital Accord*, http://www.bis.org, 2003.

16. Hand, D. J. and W. E. Henley, "Statistical Classification Methods in Consumer Credit Scoring: A Review", *Journal of Royal Statistical Society. Series A (Statics in Society),* Vol. 160, No. 3, pp. 523–541, 1997.

17. Saretto, A. A., *Predicting and Pricing the Probability of Default*, http://personal.anderson.ucla.edu/alessio.saretto/default.pdf, 2004.

18. Altman, E. I., "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy", *The Journal of Finance* Vol. 23, No. 4, pp. 589–609, 1968.

19. Standard & Poor's, *Rating Scale Services for Internal Rating Systems*, http://www2.standardandpoors.com/spf/pdf/products/RatingScaleServices.pdf, 2008.

20. Basel Committee on Banking Supervision, *International Convergence of Capital Measurement and Capital Standards*, http://www.bis.org, 2004.

21. Statlog Project Databases, *Australian Dataset*, ftp://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/australian.

22. Statlog Project Databases, *German Dataset*, ftp://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/german.

23. Wikipedia, *Hidden Markov Model*, http://en.wikipedia.org/wiki/Hidden_markov_model.

24. Rabiner, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE,* Vol. 77, No. 2, pp. 257-286, 1989.

25. Baum, L. E. and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains", *Ann. Math. Stat.,* Vol. 37, pp. 1554-1563, 1966.

26. Baum, L. E., T. Petrie, G. Soules and N. Weiss, "A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains", *Ann. Math. Stat.,* Vol. 41, pp. 164-171, 1970.

27. Viterbi, A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm", *IEEE Trans. Informat. Theory,* Vol. IT-13, pp. 260-269, 1967.

28. Forney, G. D., "The Viterbi Algorithm", *Proc. IEEE,* Vol. 61, pp. 268-278, 1973.

29. Dempster, A. P., N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algortihm", *J. Roy. Stat. Soc.,* Vol. 39, No. 1 pp. 1-38, 1977.

30. Wikipedia, *Logistic Regression*, http://en.wikipedia.org/wiki/Logistic_Regression.

31. Mojsilović, A., *A Logistic Regression Model for Small Sample Classification Problems with Hidden Variables and Non-Linear Relationships: An Application in Business Analytics*, ICASSP, 2005.

32. Xu, L, Chow, M.-C, Gao, X.Z, "Comparisons of logistic regression and artificial neural network on power distribution systems fault cause identification"*, IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications, SMCia/05*, Espoo, pp. 128 – 131, 28-30 June 2005.

33. Pezzullo, J.C., *Logistic Regression Calculating Page*, http://statpages.org/logistic.html.

34. Wikipedia, *k-Nearest Neighbors*, http://en.wikipedia.org/wiki/K_nearest_neighbors.

35. Li J., Li X. and Yao X., "*Cost Sensitive Classification with Genetic Programming*", http://www.cs.bham.ac.uk/~xin/papers/LiLiYaoCEC05.pdf, CERCIA, The University of Birmingham, 2005.

36. Wikipedia, *False Negative*, http://en.wikipedia.org/wiki/False_negative#Type_II_error.

37. MIT School of Management, *Data Mining - Spring 2003 lecture notes*, http://ocw.mit.edu/NR/rdonlyres/Sloan-School-of-Management/15-062Data-MiningSpring2003/20BB4551-FF7C-4D63-B657-4FF096F37988/0/Lecture2notes.pdf, Massachussets, 2003.

38. Wikipedia, *ROC Curve,* http://en.wikipedia.org/wiki/ Receiver_operating_characteristic_curve.

39. Hu, W., Q. Song, and W. Xie, "Robust Support Vector Machine with Bullet Hole Image Classification", *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews,* Vol. 32, No. 4, November 2002.

40. Geisler, W., J. Ghosh and T. Kuyel, "Fast Image Classification Using a Sequence of Visual Fixations", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics,* Vol. 29, No. 2, April 1999.

41. Härdle, W., R. A. Moro and D. Schäfer, *Rating Companies – A Support Vector Machine Alternative*, Bundesbank, November 2005.

42. Laan, N.C., D. F. Pace, H. Shatkay, "Initial Model Selection for the Baum-Welch Algorithm as Applied to HMMs of DNA sequences", *First Canadian Student Conference on Biomedical Computing*, School of Computing, Queen's University, Kingston, 2006.

43. University of Cambridge, *HTK Speech Recognition Toolkit*, , http://htk.eng.cam.ac.uk.