NON-LINEAR QUESTION DIFFICULTY MODELING IN COMPUTER ADAPTIVE TESTS

by

Zahraa M. Abdulhussein

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2018

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Zahraa M. Abdulhussein

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Volkan for the continuous support, for his patience, motivation, and great knowledge. His guidance helped me in all the time of research and writing of this thesis.

Most importantly, I would like to thank my parents for always supporting me throughout writing this thesis. Without your continued support, I could never have accomplished so much.

I would also like to thank my friends, Alla and Hassan. I appreciate the positive impact you have had on my life. I am grateful for your help, kindness and generosity.

ABSTRACT

NON-LINEAR QUESTION DIFFICULTY MODELING IN COMPUTER ADAPTIVE TESTS

Abdulhussein, Zahraa,

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Assoc. Prof. Dr. Oğuz Bayat

Co-Supervisor: Dr. Volkan Sevindik

Date: June 2018

Pages: 56

Computer adaptive test (CAT) designed to estimate the ability of a student, where a collection of questions that has different level of difficulty are selected according to the level of student's ability. CAT used item response theory (IRT) as a model to select the proper questions and to estimate the ability of the student. Aiming at the problems in estimation of the ability of the student in IRT, we suggest two methods which are, random walk theory and hidden Markov model (HMM). We conduct a simulation experiments for one-dimensional random walk models and HMM model. The result shows that the one-dimensional random walk models and HMM model are less complex and better than IRT model.

Keywords: Computer Adaptive Test, HMM, Random Walks, Item Response Theory.

ÖZET

BİLGİSAYARDA DOĞRUSAL OLMAYAN DENKLEM MODELLEMESİ BİLGİSAYAR ORTAMLI SINAMA

Abdulhussein, Zahraa,

Yüksek Lisans, Elektrik ve Bilgisayar Mühendisliği, Altınbaş Üniversitesi,

Danışman: Doç. Dr. Dr. Oğuz Bayat

Yardımcı Danışman: Dr. Volkan Sevindik

Tarih: Haziran, 2018

Sayfalar: 56

Öğrencinin yeteneğini tahmin etmek için tasarlanan Bilgisayar Uyarlamalı Test (CAT), öğrencinin kabiliyet düzeyine göre farklı zorluk derecesine sahip soruların bir araya getirildiği durumlarda seçilir. Bilgisayar Uyarlamalı Test, uygun soruları seçmek ve öğrencinin yeteneğini tahmin etmek için bir model olarak Öğe Yanıt Teorisi'ni (IRT) kullanmıştır. Öğrencinin Madde Tepki Teorisindeki kabiliyetini tahmin etmedeki problemleri hedefleyerek, Tesadüfî Hareket Teorisi (RWT) ve Saklı Markov Modeli (HMM) olmak üzere iki yöntem önermekteyiz. Tek boyutlu tesadüfi hareket modelleri ve saklı Markov modeli için simülasyon deneyleri yapıyoruz. Sonuç, tek boyutlu tesadüfi hareket modellerinin ve saklı Markov modelinin, Öğe Yanıt Teorisi modelinden daha az karmaşık ve daha iyi olduğunu göstermektedir.

Anahtar Kelimeler: Bilgisayar uyarlamalı testi, gizli markov modeli, rastgele yürüyüşler, madde cevap teorisi.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

1. INTRODUCTION

1.1 COMPUTER ADAPTIVE TEST

One of the most important subject in the education system is the student's assessment. Assessment helps to improve the learning process of the students and provides feedback to instructors. It can detect students' weaknesses so that they can focus on strengthen their ability to improve their education. Students may learn a subject easily however, find their weaknesses can be a little bit tricky. We know from the genetic perspective students have different levels of ability, therefore assessment can be an opportunity to identify knowledge gaps for each student. Yet, educational assessment mostly does not focus on students individually because they usually take classic tests, such as paper and pencil tests and these tests can not evaluate students' performance at the individual level.

In general, classic tests have fixed set of questions and they are written for a group of students. Which means all the students get the same questions that do not fit their abilities. Some of students find the questions too easy and some of them find the questions too hard. Which means we cannot estimate the real ability for each student. Thus, paper and pencil tests can be a waste of time and effort.

In contrast, computer adaptive test concentrates on each student. Every student gets a collection of questions that adapt to the student's ability. Firstly, a student gets a question, if he answers the question correctly then the next question will be slightly harder; if he answers the question wrongly then the next question will be slightly easier. Secondly, it estimates the ability of the student simultaneously while the student gets the questions. Finally, the process continues until the test is terminated. All these steps are implemented into a software, known as computer adaptive testing application (CAT). CAT consists of an item bank, which is where the questions are stored and addressed according to difficulty. An algorithm for selecting proper questions who depends on student's ability. Methods to calculate the ability and conditions to terminate the test. Usually, questions are multiple-choice, there is only one correct answer. Which means it is a dichotomous test that may get correct or incorrect responses from the student [1]. Furthermore, CAT is a flexible environment that has many advantages. It takes less time than the classic tests. It is accurate for estimating ability. It can submit the results after the test is finished. At any time and location, the test can be taken. And we can add feedbacks, hints or multimedia to the test which makes the exam formative and well-defined [2] and [3].

Many institutions and educational testing service have been using computer adaptive test applications. For instance, ETS organization has been employing CAT in their graduate record examination (GRE) test. It is a general test that helps to select graduate students in the United States. Also, The Graduate Management Admission Test (GMAT) who evaluates student outcome in graduate business programs and the Test of English as a Foreign Language (TOEFL) which is a standardized test for English language to evaluate non-native speakers [3] and [4].

1.2 SCOPE OF THESIS

The aim of this thesis is to estimate the ability of test taker in computer adaptive test (CAT). In general, computer adaptive test uses item response theory to estimate the ability of the student. However, item response theory has its limitations. Thus, we proposed two approaches to replace the old theory with more efficient methods.

Chapter 2, we explore computer adaptive test in depth and explain its algorithm. We discuss the theory behind CAT which is item response theory (IRT) and the advantage and disadvantage of the theory. In chapter 3, we describe first approach has been used in this thesis. Which is random walk theory and we explain how to model a one-dimensional walk. Chapter 4, describes the second approach which is Hidden Markov Model (HMM). We show the tasks that which can be solved by using HMM and explain all the calculations of the model and its applications.

Finally, Chapter 5 employs random walk theory and HMM in computer adaptive test by modeling the relationship between the difficulty of the question and the ability of the test taker.

2. COMPUTER ADAPTIVE TESTING

2.1 LITERATURE REVIEW

The history of adaptive testing began in 1905 where Alfred Binet designed the first intelligence test. He was interested in individual's ability instead of group of examinees. His method was to evaluate a set of questions according to difficulty and divided them into subsets. Then he used these questions subsets to test an examinee. If the examinee answers a subset of questions correctly then the next subset will have more difficult questions. And if the examine answers the questions wrongly then the next subset will have less difficult questions. After that Binet could estimate the examinee's ability [5].

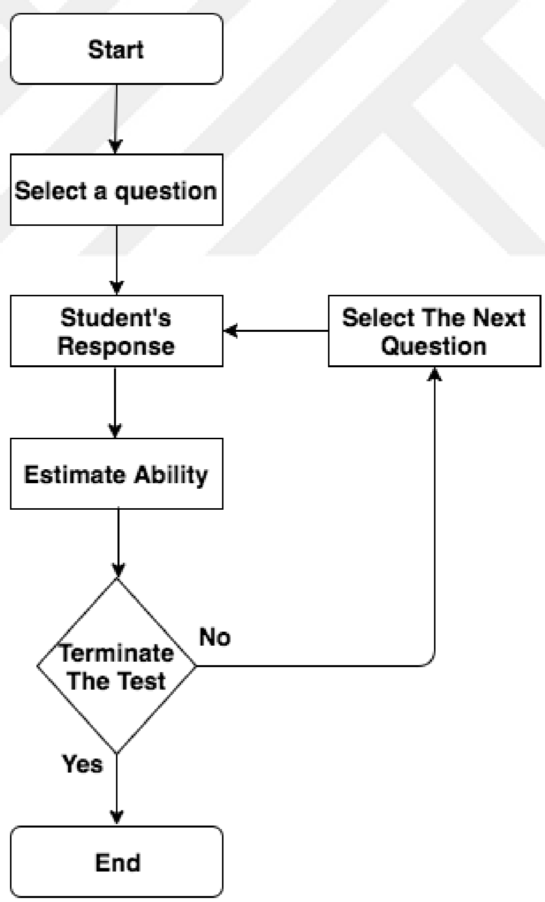

Figure 2.1: The Flowchart of Computer Adaptive Test.

Moreover, David Weiss in 1973 was the first one who employed computers to simulate adaptive testing. Which is where computerized adaptive testing began. He adopted Binet's method to estimate individual's ability. However, a computer estimates the examinee's ability instead of the examiner. There are many questions stored in the computer and ready to display to examinee according to difficulty. Until the program estimates the ability and terminates the test [6]. Also, the Self-Scoring flexilevel test by Lord in 1980, step procedure by Henning in 1987 and Testlets by Lewis and Sheehan in 1990 were similar to Binet's method but with different techniques [5].

Currently, most of CAT applications use Item response theory (IRT) as an essential model to select the probable questions, estimates the ability and evaluates the accuracy of the test. There are many IRT models, as will be discussed in more detail in section 2.2.1.

2.2 ITEM RESPONSE THEORY

The most popular theory behind computer adaptive test is item response theory (IRT). It is a mathematical approach to illustrate the relationship between the question's difficulty and examinee's ability. It used to estimate the examinee's ability (defines as the latent trait) after answering a set of questions which addressed according to difficulty.

The components of item response theory are: -

- Item response function (IRF), who calculates the probability of answering a question correctly. Such as, the one parameter logistic model. Also, it is called Rasch model, as shown in equation 2.1. Where e is the natural logarithm, approximately equal to 2.7182. Theta θ is the ability level of the student and b is the difficulty parameter. There are many IRT models, as will be discussed in the next in section.

$$p(\theta) = \frac{1}{1 + e^{-(\theta - b)}} \tag{2.1}$$

- Item Information Function, who defines how well an item to estimate the ability and it used to select each question in the exam, as shown in the equation 2.2. Where $P_i(\theta)$ is the probability of answering a question correctly for a specific item and $P_i(\theta)'$ is the first derivative of $P_i(\theta)$.

$$I_i(\theta) = \frac{P_i(\theta)'}{P_i(\theta) * (1 - P_i(\theta))} \tag{2.2}$$

- Statistical methods to estimate the ability (θ), such as, maximum-likelihood estimator (MLE), Maximum A Posteriori estimator (MAP) and the Expected A Posteriori estimator (EAP) [3].

- Test information function is used to calculate ability estimation error as condition to terminate the test. As shown in the equation 2.3, where $I(\theta)$ is the test information function and SE is the estimation standard error [7].

$$SE(\theta) = \frac{1}{\sqrt{I(\theta)}} \tag{2.3}$$

2.2.1 Item Response Theory Models

Item Characteristic Curve model (ICC) known as Item response function (IRF). It is a logistic function that calculates the probability of correct response. it has S shape curve and there are many ICC models, such as, the one-parameter logistic model known as Rasch model, as shown in equation 2.1. Which has one parameter and it is the difficulty of an item. There is another model which is the two-parameter logistic model and it has the difficulty parameter and the discrimination parameter who defines how closely an item (a question) to distinguish between a group of students near an ability level. As shown in the equation 2.4 where the discrimination parameter is denoted by a.

$$p(\theta) = \frac{1}{1 + e^{-a(\theta - b)}} \tag{2.4}$$

Also, there is the three-parameter model that has difficulty, discrimination and guessing parameters. Guessing parameter is the probability of guessing a correct answer [8], the mathematical formula for three-parameter model is: -

$$p(\theta) = c + (1 - c)\frac{1}{1 + e^{-a(\theta - b)}} \tag{2.5}$$

Where e is the natural logarithm and it is 2.7182. θ is the ability level of the student, known as the latent trait. We can not observe it directly but we can measure it by a scale that has a midpoint of zero and unit of measurement of one. Therefore, it is interval scale from negative infinity to positive infinity, but in practice, it will be from negative three to positive three. In Three-parameter logistic model at each level of ability, there is a specific probability of correct response, the highest probability of correct response goes with the highest ability and the lowest probability of correct response goes with the lowest ability.

a is the discrimination parameter, the purpose of this parameter is to distinguish high ability examinees from low ability examinees. Which means the discrimination parameter represents the slop of the probability curve, as shown in figure 2.2, the slop of the curves increases when discrimination parameter increases. If the value of a is equal to zero, then the curve will be a horizontal line and the value of the probability will be 0.5.

b is the difficulty parameter and its value set on the ability scale where the probability of correct response is in the middle, between the lowest value of the probability of correct response and 1. Also, it represents as location index to describe where the item has a probability of correct response greater or equal to 0.5. b has the same scale as ability parameter and it is from negative infinity to positive infinity. If a student has ability = 2 and the test show him a question that has b <= 2 then he may answer the question correctly with 0.5 probability or more. And if b > 2 then the probability of correct response is less than 0.5. Therefore, if the value of b is greater than the value of the ability then there is less probability of answer the question correctly. As shown in figure 2.3 there are 9 items who have different level of difficulty.

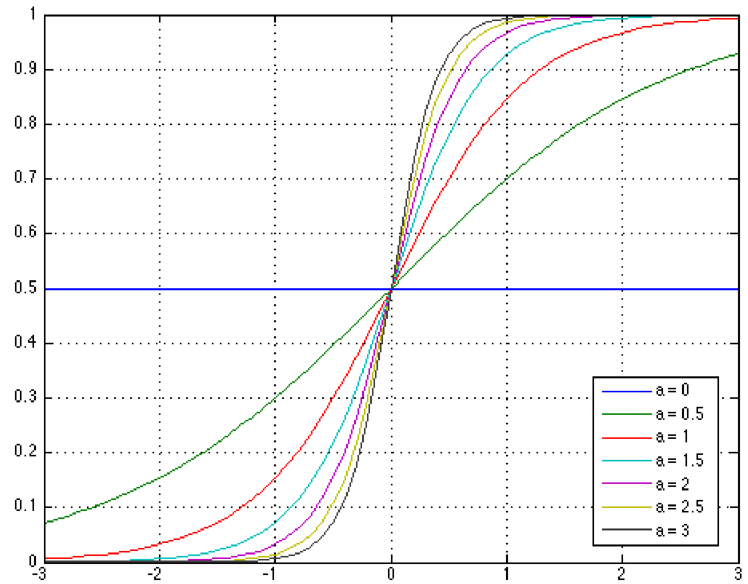

Figure 2.2: Item Characteristic Curve – Three Parameter Model.

c is the guessing parameter and its value represent the probability of solving an item correctly by guessing alone. Guessing parameter is a probability, therefore, the theoretical range is from zero to one, but in practice, it will be 0 to 0.35. Also, there is linear relationship between guessing parameter and the probability of correct response. The probability increases when guessing parameter increases. As you can see in the figure 2.4, there are five values of guessing parameter [9].

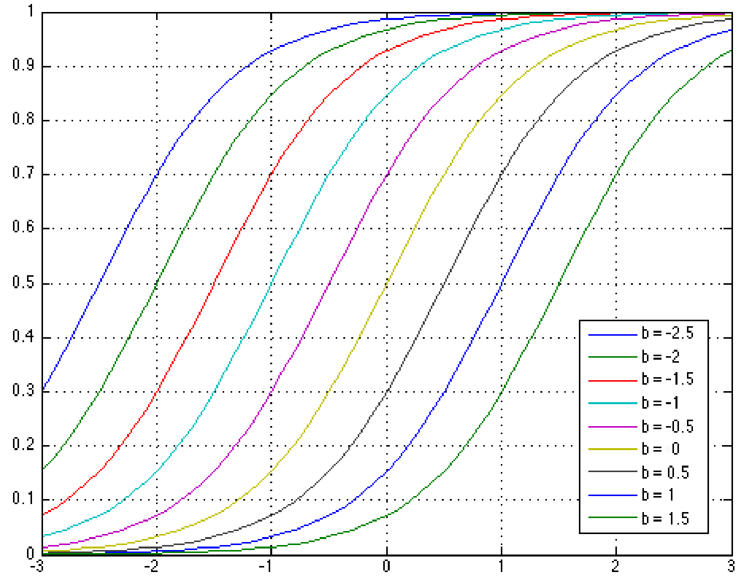

Figure 2.3: ICC Function With 9 Different Level of Difficulty. The Discrimination is 1 And Guessing Parameter is 0.

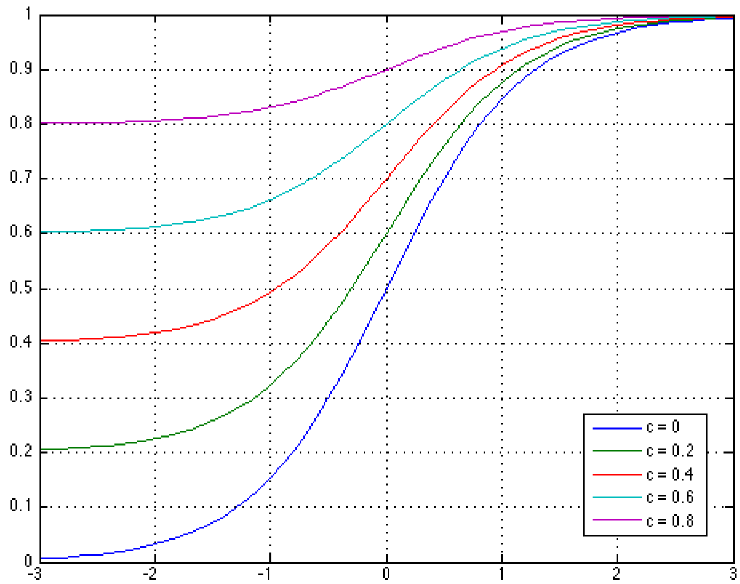

Figure 2.4: ICC Function with Five Items That Have Different Level of Guessing.

2.2.2 Advantages and Disadvantages of Item Response Theory

The advantages of IRT are:

- Exam time is shorter than the classic test as a result of taking less number of questions to estimate the ability.
- IRT provides accurate estimation of the ability.
- The examinees are not bored while taking the exam because IRT selects the optimal item whose difficulty is near the ability of the student. Which means the student always gets a proper question and the selection process adjust according to the ability of the student [7].

The disadvantages of IRT are:

- When an examinee answers all the questions correctly, then IRT estimates the ability as positive infinity. And when all the answers are wrong, it estimates the ability as negative infinity. Which means in these cases IRT fails to estimate the ability [9].
- Sometimes IRT selects certain items more than once and neglect the others, which means that some of the items are being exposed too much in the test while other items are not being used.
- IRT algorithm does not know from which topic is the question, which means it may select many items from one topic and neglect the other topics.
- In order to estimate the parameters of the items accurately, IRT requires careful Item calibration and real data measurements which means it needs effort to collect the data that may not easy to obtain [7].

3. RANDOM WALK THEORY

3.1 WHAT IS RANDOM WALK THEORY?

A random walk is a change that happens without any pattern. It is a stochastic process which consists of a sequence of steps where the steps are heading to random directions. The direction of each step is set by a probability that used to estimate the occurrence of an event after taking a sequence of random steps. There are different types of random walk processes. Some of random walks has an assumption that the current step does not depend on the previous step whereas for other random walks', current step depends on the pervious step.

The size of the steps may be fixed-length or various-length, number of the steps may be finite or infinite, and the process can be continuous-time random walk or discrete-time random walk. Moreover, the space of random walk may be n-dimensional for n directions, such as Euclidean space which means it can be one-dimensional that has only two directions, left and right or forward and backward, figure 3.1 illustrates one-dimensional walker moved 6 steps, 2 to the right and 4 to the left and it ends up at position -2. Also, there are two-dimensional random walk, each point has four neighbors and three-dimensional space where each point has six neighbors [10].

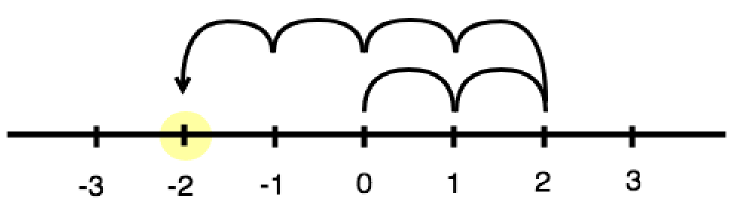

Figure 3.1: One-Dimensional Walker

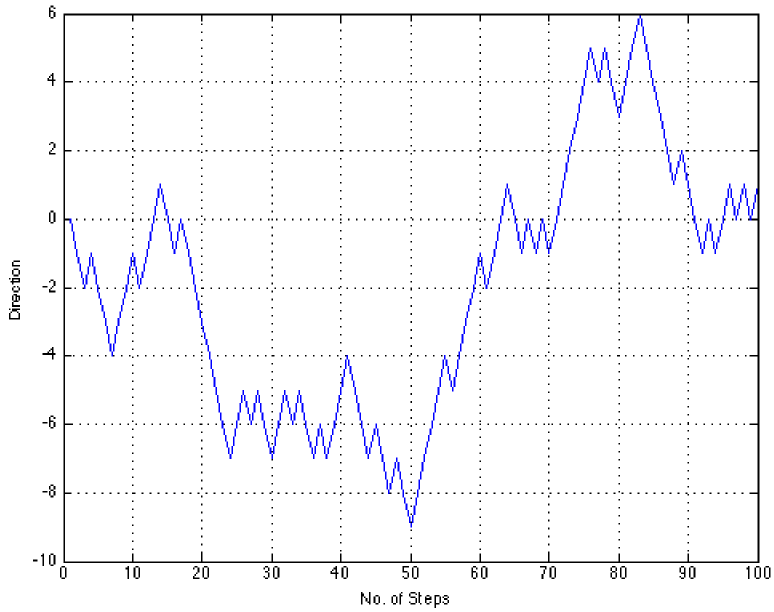

Figure 3.2: One-Dimensional Walker Took 100 Steps and Settled Down at Position 1.

3.2 THE CONSTRUCTION OF RANDOM WALK THEORY?

3.2.1 One-Dimensional Walks

Suppose there is a man “a walker’ and we observe his movement in discrete space with a step size of 1. We assume that he starts at position zero and he flips a coin to select his next step. If the coin comes with head then he moves to right position with a step size of 1; if the coin comes with a tail then he moves to the left position with a step size of 1 and this process continues. Thus, there is 50% chance that the walker moves to the left, there is 50% chance that the walker moves to the right and there is zero chance that walker stays at the same position. Therefore, the probability of going right, denoted by p, is 0.5 and similarly, the probability of going left, denoted by q=1-p, is 0.5. The total number of steps that the walker moved randomly will be denoted by N. For now, let’s assume that a total number of 5 steps have been made by the walker, that is, 3 steps to the right, denoted by n1, and 2 steps to the left, denoted by n2. By using combination rule in probability theory, we can find out how many ways the walker can make these five steps, regardless of what all the pervious steps are. This can be calculated as below [10] and [34]: -

$$\binom{N}{n_1} = \frac{N!}{n_1!\,(N-n_1)!} = \frac{N!}{n_1!\,n_2!}$$

$$\binom{5}{3} = \frac{5!}{3!\,(5-3)!} = \frac{5!}{3!\,2!} = 10 \tag{3.1}$$

where N! denotes factorial in equation 3.1

There are 10 ways that walker can take 5 steps with 3 steps towards to a particular direction. This number is binomial coefficient. Thus, we can use binomial distribution to estimate the probability of moving these five steps given that the man moved 3 steps to right already: -

$$P(n_1) = \frac{N!}{n_1!\,n_2!} * p^{n_1} * q^{n_2} \tag{3.2}$$

$$P(3) = \frac{5!}{3!\,2!} * (0.5)^3 * (0.5)^2 = \frac{10}{32}$$

Therefore, in general we estimate the probability of a particular sequence of N steps or the probability of a position after taking N steps by using binomial distribution. Table 3.1 shows the probabilities of one to five steps for one dimensional unbiased discrete random walks [34].

Table 3.1: Probability Distribution for One-Dimensional Discrete Random Walk.

No. of steps (N)	Walker's position										
	-5	-4	-3	-2	-1	0	1	2	3	4	5
0						1					
1					$^1/_2$	0	$^1/_2$				
2				$^1/_4$	0	$^2/_4$	0	$^1/_4$			
3			$^1/_8$	0	$^3/_8$	0	$^3/_8$	0	$^1/_8$		
4		$^1/_{16}$	0	$^4/_{16}$	0	$^6/_{16}$	0	$^4/_{16}$	0	$^1/_{16}$	
5	$^1/_{32}$	0	$^5/_{32}$	0	$^{10}/_{32}$	0	$^{10}/_{32}$	0	$^5/_{32}$	0	$^1/_{32}$

As can be seen from the table above, according to the binomial distribution the walker may end up at position 1 or -1 with $10 / 32$, in order to find his exact position or the distance d that the walker has reached, we should use this equation below [10]: -

$$d = n_1 - n_2 \tag{3.3}$$

$$= 2\, n_1 - N$$

From the equation 3.3, the man will always end up at position 1, regardless of the order of the directions, as shown in Figure 3.3, all the possible direction combinations of the man may take. Where R means right direction and L means left direction.

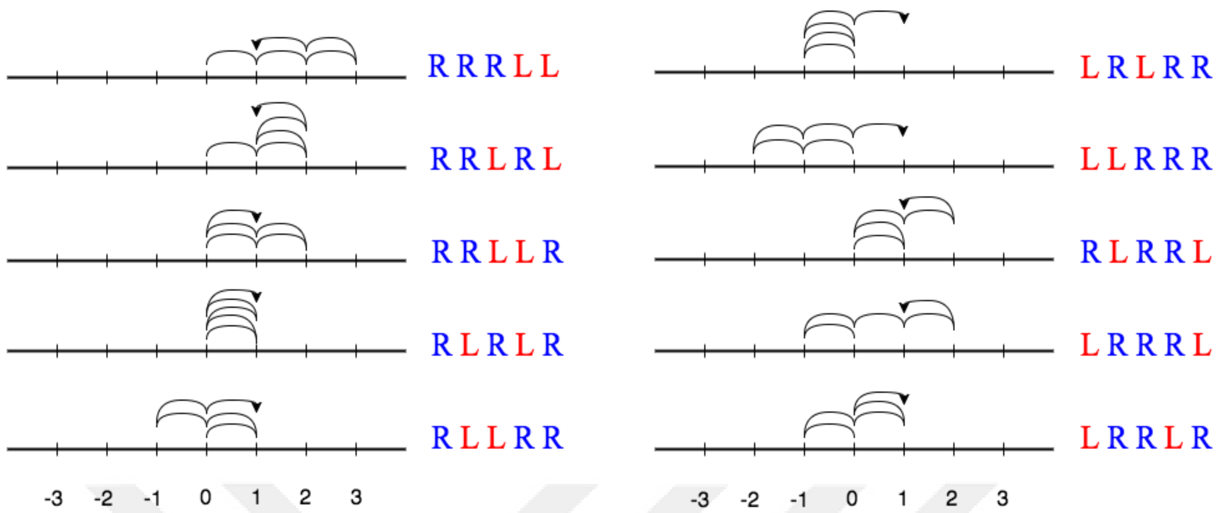

Figure 3.3: One-Dimensional Walker Took Five Steps, 2 To Left And 3 To the Right Direction.

The mean and variance for the right steps are: -

$$\mu_{n_1} = N\,p \quad and \quad \sigma^2{}_{n_1} = N\,p\,q \tag{3.4}$$

Where, the mean of the left steps is: -

$$\mu_{n_2} = N\,q \tag{3.5}$$

3.3 APPLICATIONS OF RANDOM WALK THEORY

There are several areas where Random walk theory has been employed, these are: -

- In financial economics, the stock market prices follow random walk, used to predict and analyze the behavior of stock markets and currency exchange rates where prices change randomly and independently [11] and [12].
- In psychics, molecules collide with particles of liquid or gas in random directions, this process is known as Brownian motion which is an example of a random walk [10].

- In Biology, random walk theory model bacteria movements where bacterial cells swim randomly in a chemical environment [13].
- In computer science, random walk theory used as algorithm for recommendation services in social network [14].

  Also, is used in image segmentation where random walk model predicts the contents of an image [15] and mining very large graphs in complex networks where random walk algorithm estimates motif statistics [16].

4. HIDDEN MARKOV MODEL

4.1 WHAT IS HIDDEN MARKOV MODEL?

Hidden Markov model or HMM is a temporal probabilistic model based on the concept of a Markov chain. HMM predicts a sequence of states or events that develop over time where the states are hidden. There are a few observations that lead us to the states prediction with an assumption that the current state depends only on the previous state. For instance, suppose we want to predict the weather whether it is sunny or rainy, and assume that we can not observe the weather directly. The state of the weather is hidden from us but we have some evidences such as person's mood. We observe if the person is happy or grumpy and from these observations we can predict the weather state [17]. There are several tasks can be solved by using HMM, these are:

- **Filtering**, to predict the current state given the observations of the previews state, known as state estimation.
- **Prediction**, to estimate the future state given the observations of the past, it is useful to estimate the expected outcome of an event.
- **Smoothing**, to predict the previous state given the observations of the current state, this task optimizes the estimation of the past state compared to recent one [17].

Also, there is **learning** task where we have a sequence of states and we want to estimate the parameters of HMM. Learning problem can be solved by using Baum-Welch algorithm as shown in section 4.1.4.

In order to understand hidden Markov model, we need to understand the concept of Markov chain, as will be discussed in more detail in the next section.

4.1.1 Markov Chain

Markov chain or Markov process is a random process that characterizes a system whose state develop over time with an assumption that the current state depends only on the previous state and each state has a probability which changes over time according to assigned probabilities known as transition probabilities [18]. Markov chain applied in several applications, such as climate prediction [19], analyze throughput and improve network management [20] [21], and probabilistic analysis of the flow of wind power [22].

By way of illustration, let us assume we have two weather events, sunny and rainy. Based on collected data, there is 60% chance that rainy day will stay rainy in the next day and there is 80% chance that sunny day will stay sunny in the next day. These probabilities are called transition probabilities as shown in figure 4.1. Suppose the initial state for rainy is one which means the initial state for sunny is zero. By using product rule in probability theory, we can calculate the probability of rainy P(Rainy1) and sunny P(Sunny1) in next state or next day [18]:

$$P(Rainy1) = P(Rainy0) \times P(Rainy|Rainy) + P(Sunny0) \times P(Rainy|Sunny)$$

$$P(Sunny1) = 1 - P(Rainy1)$$

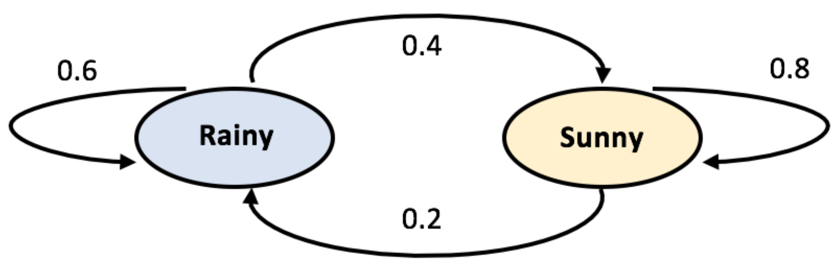

Figure 4.1: Transition Probabilities for Two Hidden States.

Therefore, P(Rainy1) is 0.6 whereas the probability of sunny P(Sunny1) is 0.4. To estimate the probability for the second day we will use the probability of the first day as the initial state and so on for all states because of Markov chain property, each state depends only on the previews state, therefore: -

$$P\ (Rainy2)\ =\ 0.6\ *\ 0.6\ +\ 0.4\ *\ 0.2\ =\ 0.44$$

$$P\ (Sunny2)\ =\ 1\ -\ P\ (Rainy2)\ =\ 0.56$$

and there is another way to solve it by using matrix. We put the transition probabilities in a matrix and the initial state in vector and multiply by each other, as shown below: -

$$P[\ Rainy2\quad Sunny2] = [\ Rainy1\quad Sunny1\ ] \,.\begin{bmatrix} Rainy|Rainy & Sunny|Rainy \\ Rainy|Sunny & Sunny|Sunny \end{bmatrix}$$

$$P[\ Rainy2\quad Sunny2] = [\ 0.6\quad 0.4\ ] \,.\begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix} = [0.44\quad 0.56]$$

4.1.2 Estimating the Probability of a Hidden State

The difference between HMM and Markov chain is that in hidden Markov model we can not see the state, to be more specific, we cannot see the weather is rainy or sunny, the state is hidden from us but we have some evidences or observations that lead us to the state. We use the Bayes's theorem in the probability theory to predict the hidden state, as shown in equation 4.1. This estimation problem is known as filtering to estimate the posterior probability of a hidden state [17] and [35].

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{4.1}$$

let us go back to the weather example and assume we have initial state 50% for rainy. The observation is a person and he is either happy or grumpy. These observations have probabilities called emission probabilities and they depend on the state which we cannot see. Figure 4.2 illustrates the transition probabilities and observations probabilities for all time.

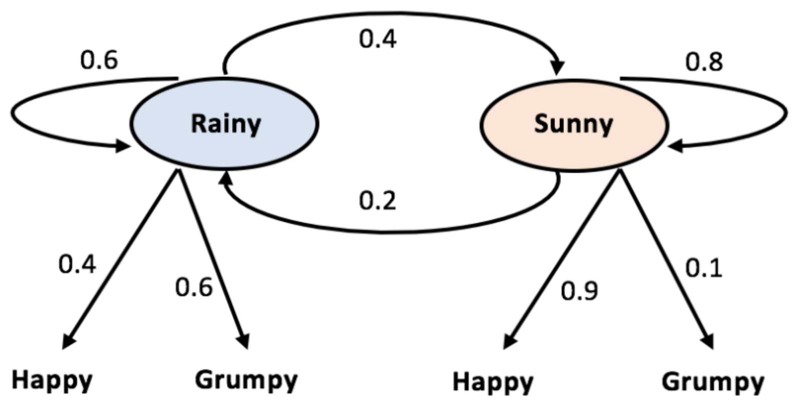

Figure 4.2: Transition Probabilities and Emission Probabilities in HMM.

Suppose on day 1 we observe that the person is happy, what is the probability of raining on day one? By using Bayes' theorem in equation 4.1, we can estimate the probability of raining on day 1 given the person is happy, therefore: -

$$P(Rainy1|Happy1) = \frac{P(Happy1|Rainy1) * P(Rainy1)}{P(Happy1)}$$

where P (Rainy1) = 0.5 * 0.6 + 0.5 * 0.2 = 0.4 thence P (Sunny1) =0.6

P (Happy1) = 0.4 * 0.4 + 0.9 * 0.6 = 0.7

P (Happy1| Rainy1) = 0.4

Therefore P(Rainy1|Happy1) is 0.22857 whereas the probability of sunny given happy P(Sunny|Happy1) is 1 – P(Rainy|Happy1) = 0.77143 and since sunny has the higher probability, then the hidden state is sunny. Figure 4.3 illustrates HMM using trellis diagram, where Sunny1 represents posterior hidden state probability for day one given initial state. Observation which is Happy1 and then depends on Sunny1 and given Grumpy2 on day 2. We can estimate posterior probability for second hidden state which is Rainy2. And this process continues till we estimate the posterior probabilities of hidden states given sequence of observations.

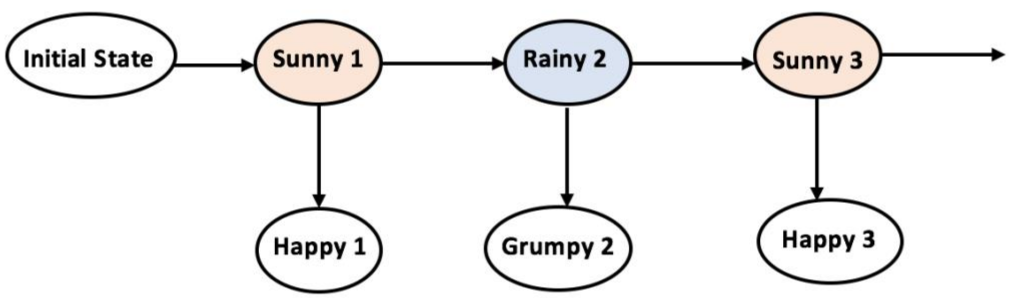

Figure 4.3: HMM Trellis Diagram.

4.1.3 The Baum-Welch Algorithm

In the 1966, Leonard E. Baum and Ted Petrie published Baum-Welch algorithm as approach to estimate the transition and emission probabilities given a sequence of observations in HMM model [23]. Which means Baum-Welch algorithm solves the learning task by estimating the parameters of HMM model.

Baum-Welch algorithm uses forward-backward algorithm and expectation maximization algorithm (EM) to train HMM model given a sequence of observation and arbitrary assumptions of the parameters which are the initial probabilities, transition probabilities and emission probabilities [24].

By way of illustration, let us back to weather example, and suppose we do not know the transition probabilities between the states which are rainy and sunny. Also, we do not know the emission probabilities and the initial probabilities. All we have is a set of observations, as shown in table 4.1 below:

Table 4.1: Sequence of Observations for HMM Model.

No	Observations
1	Grumpy, Grumpy
2	Grumpy, Grumpy
3	Grumpy, Grumpy
4	Grumpy, Grumpy
5	Grumpy, Happy
6	Happy, Happy
7	Happy, Grumpy
8	Grumpy, Grumpy
9	Grumpy, Grumpy

Firstly, we guess the initial probability for rainy and sunny, such as, 0.2 for sunny and 0.8 for rainy. Also, we guess the transition probabilities, as show in table 4.2: -

Table 4.2: Arbitrary Assumption of Transition Probabilities.

	Sunny	Rainy
Sunny	0.5	0.5
Rainy	0.3	0.7

And we guess the emission probabilities too, as shown in table 4.3: -

Table 4.3: Arbitrary Assumption of Emission Probabilities.

	Grumpy	Happy
Sunny	0.3	0.7
Rainy	0.8	0.2

Secondly, let π_i be the initial probability for each state, a_{ij} is the transition probability, $b_i(y_i)$ is the emission probability given a specific state. The forward algorithm is: -

$$\alpha_i(1) = \pi_i * b_i(y_1) \tag{4.2}$$

$$\alpha_i(t+1) = b_i(y_{t+1}) \sum_{j=1}^{N} \alpha_j(t) * a_{ji} \tag{4.3}$$

Where t is time and α_i is joint probability. We repeat this procedure for all observations. As for backward algorithm, from its name the procedure begins from the end of the sequence, as shown in the equations below: -

$$\beta_i(T) = 1 \tag{4.4}$$

$$\beta_i(t) = \sum_{j=1}^{N} \beta_i(t+1) * b_i(y_{t+1}) * a_{ij} \tag{4.5}$$

Where $\beta_i(t)$ is joint probability. By using Bayes' theorem in probability theory, we can find the highest probability after observing a sequence of evidence, as shown in the equation 4.6.

$$\gamma_i(t) = \frac{\alpha_i(t) * \beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t) * \beta_j(t)} \tag{4.6}$$

Also, we calculate the probability of state i at time t given a sequence of observations, as shown in equation 4.7,

$$\xi_{ij}(t) = \frac{\alpha_i(t) * \alpha_{ij} * \beta_i(t+1) * b_j(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t) * \alpha_{ij} * \beta_i(t+1) * b_j(y_{t+1})} \tag{4.7}$$

and then we estimate the new transition probability as show in equation 4.8.

$$a_{ij}^{*} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \tag{4.8}$$

By using all the equations above, we can calculate the transition probability from sunny to rainy given a sequence of observations in table 4.1. As shown in table below.

Observations	**Probability of Sequence and State is Sunny Then Rainy '$\boldsymbol{\gamma_i(t)}$'**	**Highest Probability Given a Specific Observing Sequence '$\boldsymbol{\xi_{ij}(t)}$'**
Grumpy, Grumpy	0.024	0.3584
Grumpy, Grumpy	0.024	0.3584
Grumpy, Grumpy	0.024	0.3584
Grumpy, Grumpy	0.024	0.3584
Grumpy, Happy	0.006	0.1344
Happy, Happy	0.014	0.0490
Happy, Grumpy	0.056	0.0896
Grumpy, Grumpy	0.024	0.3584
Grumpy, Grumpy	0.024	0.3584
Total	**0.22**	**2.4234**

Thus, the new transition probability from sunny to rainy is:

$$a^*_{sunny\ to\ rainny} = \frac{0.22}{2.4234} = 0.0908$$

We repeat the steps above to estimate all the transition probabilities and normalize them to make sure that their summation is 1. As shown in table 4.4. Finally, we repeat the forward-backward algorithm again until those probabilities converge.

Table 4.4: New Estimation of Transition Probabilities by Baum Welch Algorithm.

New Transition Probabilities		
	Sunny	Rainy
Sunny	0.0598	0.0908
Rainy	0.2179	0.9705
Normalize Transition Probabilities		
	Sunny	Rainy
Sunny	0.3973	0.6027
Rainy	0.1833	0.8167

4.2 APPLICATIONS OF HIDDEN MARKOV MODEL

There are several fields where HMM has been employed, these are:

1. Speech recognition and speech synthesis, HMM predicts specific letters in a specific word where observations are audio signals in case of speech recognition and vice versa in case of speech synthesis [25] and [26].

2. Direction estimation, HMM predicts walkers' directions for accident avoidance system in smart cities [27].

3. In astronomy, discrete HMM identifies stars where the observations sequence extracted from stars' images [28].

4. In computer networks, HMM estimates traffic and manages bandwidth in an orthogonal frequency division multiple access passive optical networks (OFDMA-PONs) [29].

5. In Biology, HMM used to classify genes [30] and to predict the mutation of suppressor genes whose cell has a high probability of provoking cancer [31].

6. In environmental studies, HMM used to model efficient automated vocalization method to detect Bryde's whale acoustic calls [32] Also, HMM with a Gamma distribution used to predict ground-level ozone in the outdoor air [33].

5. SIMULATION AND RESULTS

5.1 TEST ENVIRONMENT

We used MATLAB to simulate all the experiments and the input data we used, such as, the observations sequence in HMM model was generated by random function in MATLAB.

5.2 COMPUTER ADAPTIVE TESTING IN ONE-DIMENSIONAL RANDOM WALKS

CAT algorithm states that when we answer a question correctly, the next question will be harder than the previous question. And when we answer a question wrongly, then the next question will be easier than the previous question.

We applied CAT algorithm in random walk theory where the walk has a condition. Which is if the test takers answer a question correctly then they take a one unit step to the right direction where the harder questions will be. And when the test takers answer a question wrongly then they take one unit step to the left direction where the easier questions will be.

As shown in figure 5.1, we simulate one-dimensional walk for 100 test takers. All the test takers start from zero position where the first question shows. The green line represents the test takers who have probability of answer a question correctly is greater than or equal to 0.5, which means the test takers are heading to the right direction where the harder questions are located. Also, the blue line represents the test takers who have probability of answering a question correctly is less than 0.5. Which means the test takers are heading to the left direction where the easier questions are located.

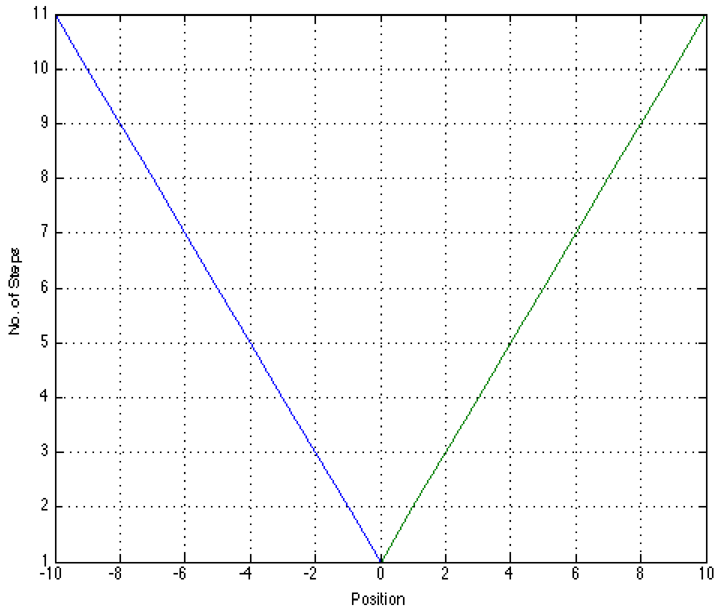

Figure 5.1: One-Dimensional Walkers in Computer Adaptive Test.

5.2.1 Ability Estimation in One-Dimensional Walk

We know in random walk theory there is a transition probability between steps which means every position has a particular probability. This probability will represent the ability level of the test taker and the position of the walker will represent the difficulty level of the question. Therefore, we created five different models to estimate the ability of the test taker. Two of them are linear models and the others are nonlinear models, as shown in table 5.1. Where the x-axis is the difficulty level of the question whose interval scale from -10 to +10. And the y-axis is the ability level of the test taker where the sum of its values is equal to 1.

We know the question with high difficulty can be solved by test taker whose ability is high. Thus, in the models the ability level increases when the difficulty level increases. As shown in table 7 the first and second equations are linear and there is slight difference in their ability levels, as shown in figures 5.2 and 5.3. The third, fourth and fifth equations are nonlinear models, as you can see from figures 5.4, 5.5 and 5.6.

Table 5.1: Five Models to Estimate the Ability in One-Dimensional Random Walk.

No.	Ability estimation models	Line chart
1	$y = \frac{10 + x}{\sum_{x=-10}^{10} (10 + x)}$	Figure 5.2
2	$y = \frac{15 + x}{\sum_{x=-10}^{10} (15 + x)}$	Figure 5.3
3	$y = \frac{1000 + x^3}{\sum_{x=-10}^{10} (1000 + x^3)}$	Figure 5.4
4	$y = \frac{1}{10.5\,[1 + \exp(-x)]}$	Figure 5.5
5	$y = \frac{1.1^x}{\sum_{x=-10}^{10} 1.1^x}$	Figure 5.6

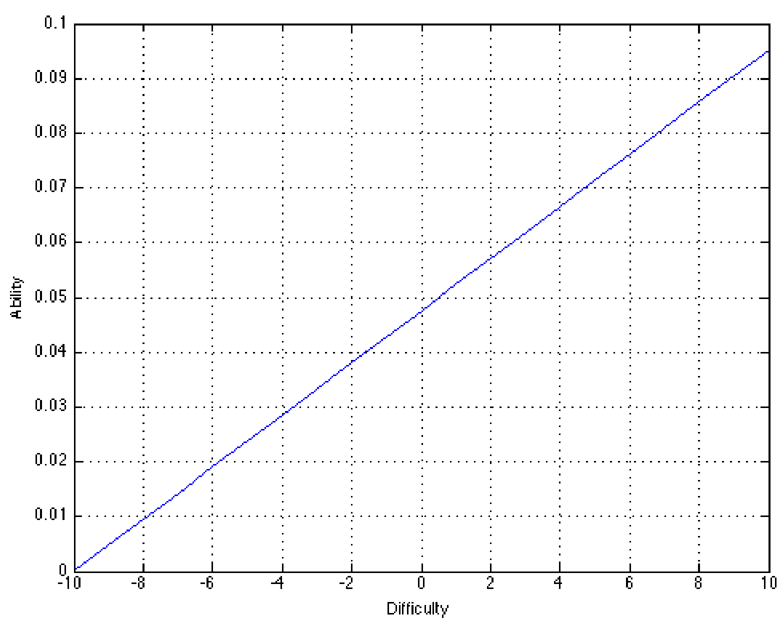

Figure 5.2: Linear Model to Estimate the Ability of The Test Taker Where the Line Intersects the Y-Axis At 0.

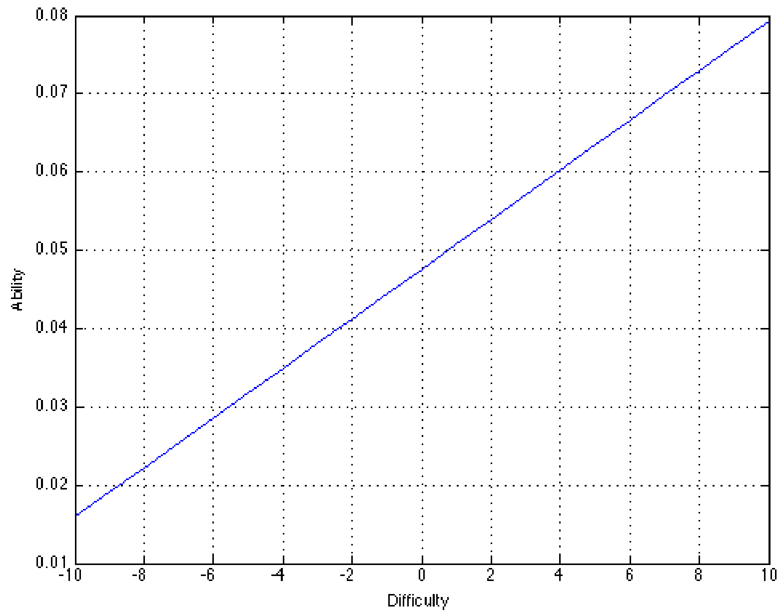

Figure 5.3: Linear Model to Estimate the Ability of The Test Taker Where the Line Intersects the Y-Axis At 0.

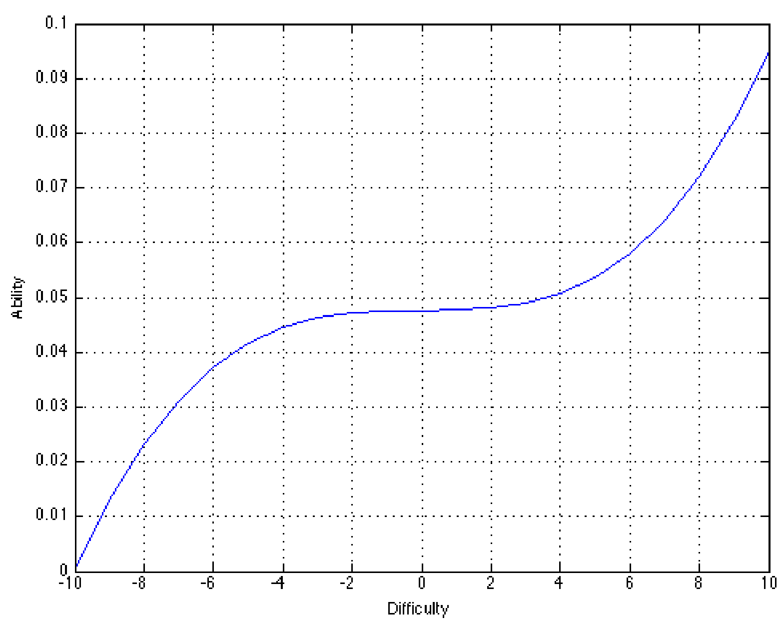

Figure 5.4: Linear Model to Estimate the Ability of The Test Taker Where the Line Intersects the Y-Axis At 0.

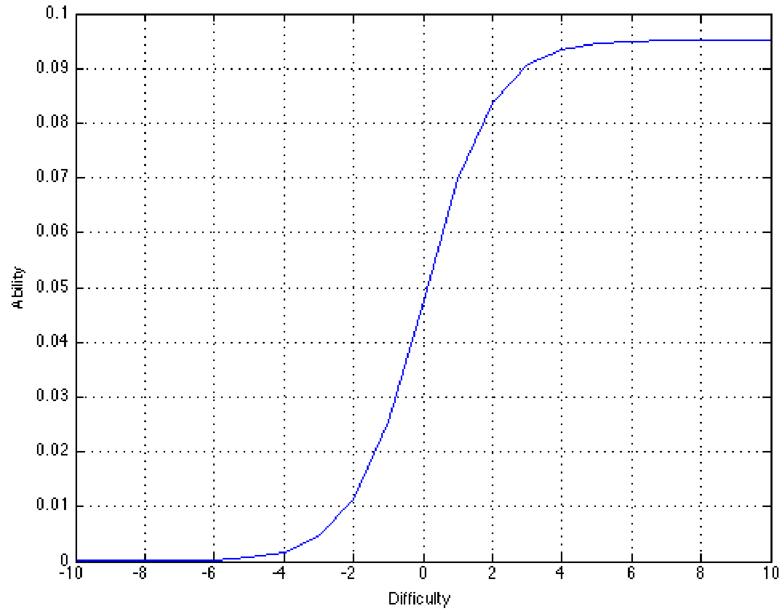

Figure 5.5: Non-Linear Model to Estimate the Ability Where the Ability Changes Rapidly Near O Position.

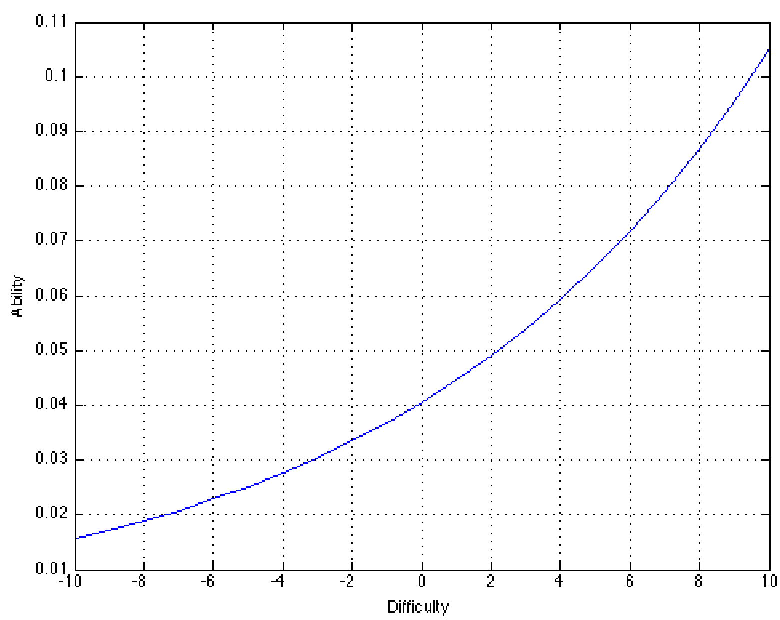

Figure 5.6: Non-Linear Model to Estimate the Ability and It Increases Gradually.

5.3 COMPUTER ADAPTIVE TESTING IN HIDDEN MARKOV MODEL

We created hidden Markov model that has 21 hidden states and they represent the difficulty levels of the questions and they are from -10 to 10, and the transition probabilities represent the ability level of a test taker, as shown in figure 5.7. We know that CAT algorithm states that when we answer a question correctly, the next question will be harder than the previous question. And when we answer a question wrongly, then the next question will be easier than the previous question. By way of illustration, suppose the test taker gets a question with 0 difficulty. If he answers the question correctly then HMM predicts the next question will have 1 level of difficulty. If the test taker answer the question wrongly then HMM predicts the next question will have -1 level of difficulty. Also, HMM predicts zero chance that the test taker gets a harder question that has 2 level of difficulty in case he answers the question correctly and there is zero chance that the test taker gets an easier question that has -2 level of difficulty in case he answers the question wrongly. We do not want to show test taker an easy question first and after that a very hard question. Also, there is zero chance that the test taker stays in the same difficulty level. Thus, every hidden state (difficulty level) has two transition probability, as shown in figure 5.7.

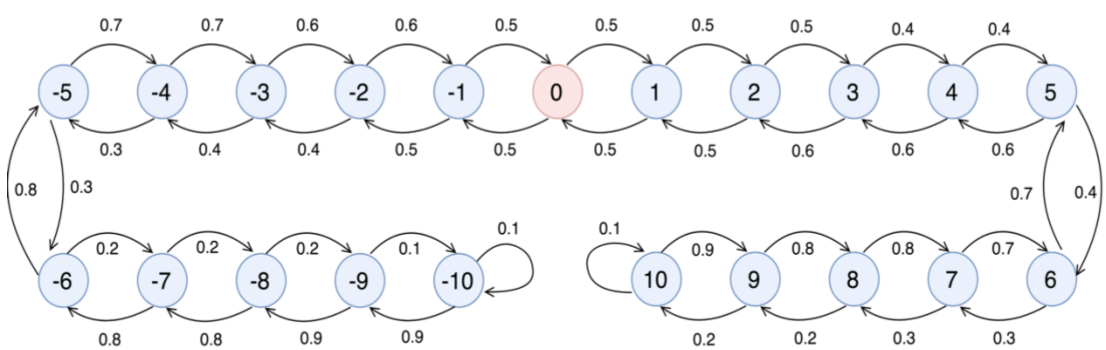

Figure 5.7: Transition Probabilities For 21 Levels of Difficulty.

For the connection between the states. The states are connected serially to each other as shown in the figure 5.7. As for the -10th and 10th states we assumed if the test taker gets a question with -10 level of difficulty and answers the question wrongly then the model shows him another question that has -10 level of difficulty. This is the same for 10th state. If the test taker gets a question with 10 level of difficulty and answers the question correctly then the model shows him another question that has 10 level of difficulty until the test is terminated.

5.3.1 Ability Estimation in Hmm

We used Baum Welch algorithm to estimate the transition probabilities, which means it estimates the ability level of a test taker given a sequence of observations, as shown in table 5.2. The observations are wrong when the test taker answers a question wrongly and right when the test taker answers a question correctly. We assumed the emission probabilities for the observations and the initial probabilities for all states, as shown in table 5.3. Also, we assumed the transition probabilities, as shown in figure 5.7. After many iterations, the algorithm estimated the ability for the test taker, as shown in table 5.4.

Table 5.2: Sequence of Observations to Estimate the Ability and It Is Generated by Random Function In MATLAB.

No.	Observations
1	wrong, wrong
2	wrong, wrong
3	right, wrong
4	wrong, wrong
5	wrong, right
6	wrong, right
7	wrong, right
8	right, wrong
9	right, wrong
10	right, right

Table 5.3: Arbitrary Assumption of Emission Probabilities and Initial Probabilities Where the Sum of Initial Probabilities Is Equal to One.

Level of difficulty	**Emission probabilities**		**Initial probabilities**
	Right	**Wrong**	
-10	0.9	0.1	1/101
-9	0.9	0.1	2/101
-8	0.9	0.1	3/101
-7	0.8	0.2	4/101
-6	0.8	0.2	5/101
-5	0.8	0.2	6/101
-4	0.8	0.2	7/101
-3	0.8	0.2	8/101
-2	0.8	0.2	9/101
-1	0.7	0.3	10/101
0	0.7	0.3	11/101
1	0.7	0.3	10/101
2	0.7	0.3	9/101
3	0.7	0.3	8/101
4	0.7	0.3	7/101
5	0.6	0.4	6/101
6	0.6	0.4	5/101
7	0.6	0.4	4/101
8	0.5	0.5	3/101
9	0.5	0.5	2/101
10	0.5	0.5	1/101

Table 5.4: The New Transition Probabilities After Normalization.

Level of Difficulty (S_i)	New Transition Probabilities		
	S_{i-1}	S_i	S_{i+1}
-10	0	0.1000	0.9000
-9	0.1000	0	0.9000
-8	0.0895	0	0.9105
-7	0.1848	0	0.8152
-6	0.2000	0	0.8000
-5	0.3000	0	0.7000
-4	0.3000	0	0.7000
-3	0.4000	0	0.6000
-2	0.3789	0	0.6211
-1	0.4821	0	0.5179
0	0.5000	0	0.5000
1	0.5000	0	0.5000
2	0.5000	0	0.5000
3	0.6000	0	0.4000
4	0.5838	0	0.4162
5	0.5870	0	0.4130
6	0.7000	0	0.3000
7	0.6891	0	0.3109
8	0.7934	0	0.2066
9	0.8000	0	0.2000
10	0.9000	0.1000	0

5.4 LINEAR AND NON-LINEAR SCORING MODELS

When the test taker finishes the exam, he should get a result, such as the number of the questions answered correctly. However, we know each question has different level of difficulty, therefore, each question gets different grading points. Such as, If the test shows the test taker a question that has 0 level of difficulty and the test taker answers the question correctly, then the test taker scores 5 points, at +1 test taker will score 7 points, at +2 level of difficulty test taker will score 10 points. Similar for the -2, -1 direction.

Thus, we created five different scoring models, two of them are linear models and the others are nonlinear models, as shown in table 5.5. Where the x-axis represents the difficulty of the question and the y-axis represents the score, as shown in figures 5.8, 5.9, 5.10, 5.11 and 5.12.

Table 5.5: Linear and Non-Linear Scoring Models.

No.	Scoring model	Line chart
1	$y = 5 + x$	Figure 5.8
2	$y = 5 + 5\,x$	Figure 5.9
3	$y = 5 + 0.2\,x^{3}$	Figure 5.10
4	$y = \frac{10}{1 + \exp(-x)}$	Figure 5.11
5	$y = 4 + 2^{\,0.25\,x}$	Figure 5.12

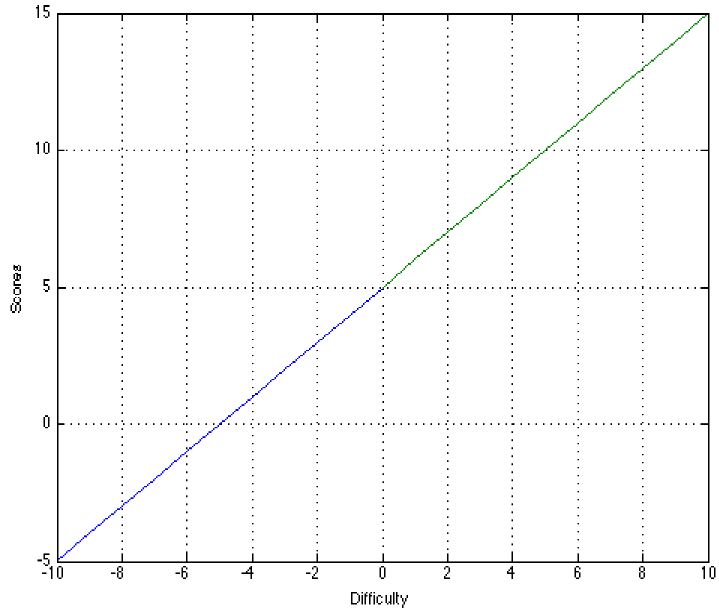

Figure 5.8: Linear Model to Calculate the Score, Where the Score Range From -5 To 15.

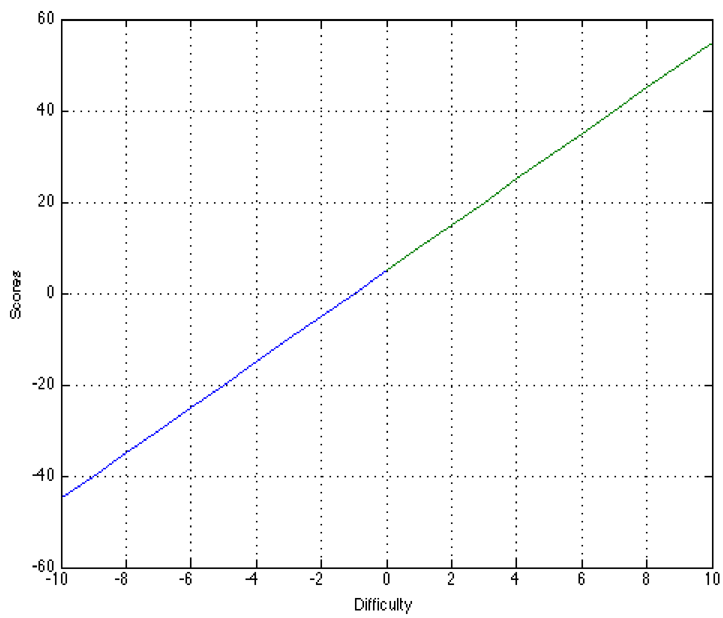

Figure 5.9: Linear Model to Calculate the Score, Where the Score Range From -45 To 55.

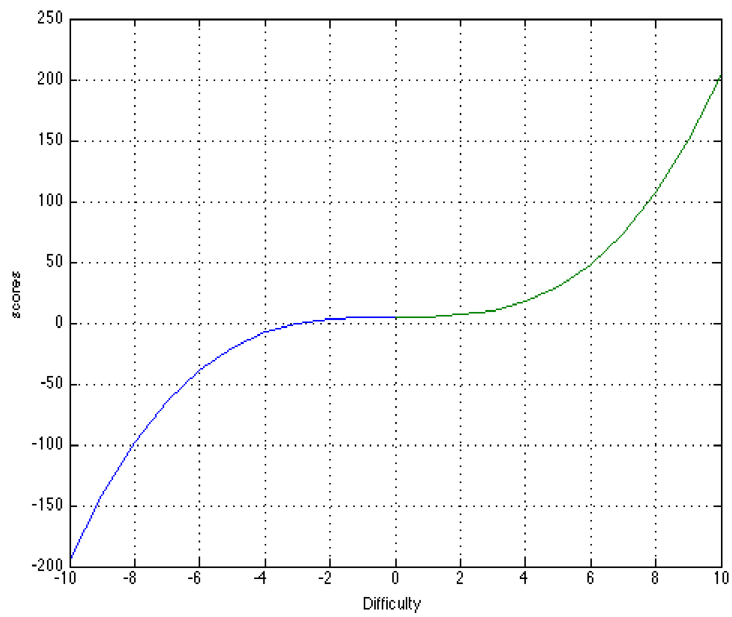

Figure 5.10: Non-Linear Model to Calculate the Score, Where the Score Range From -200 To 200.

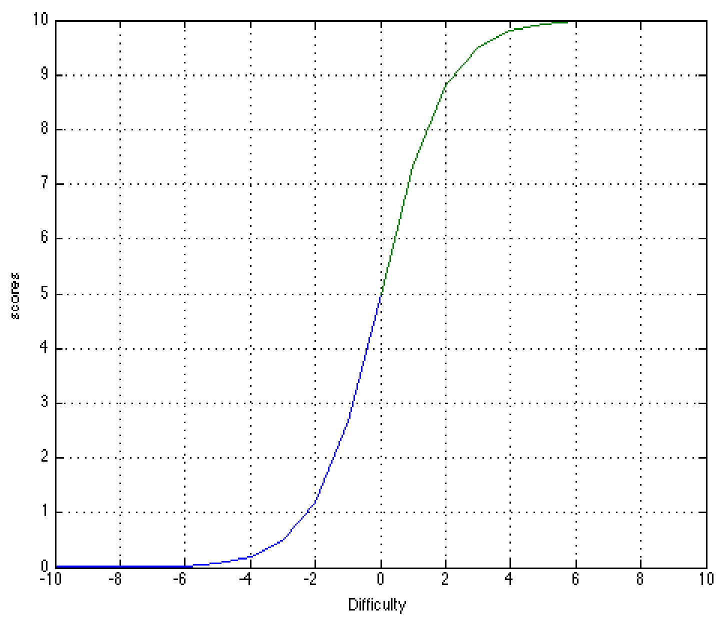

Figure 5.11: Non-Linear Model to Calculate the Score, Where the Score Range From 0 To 10.

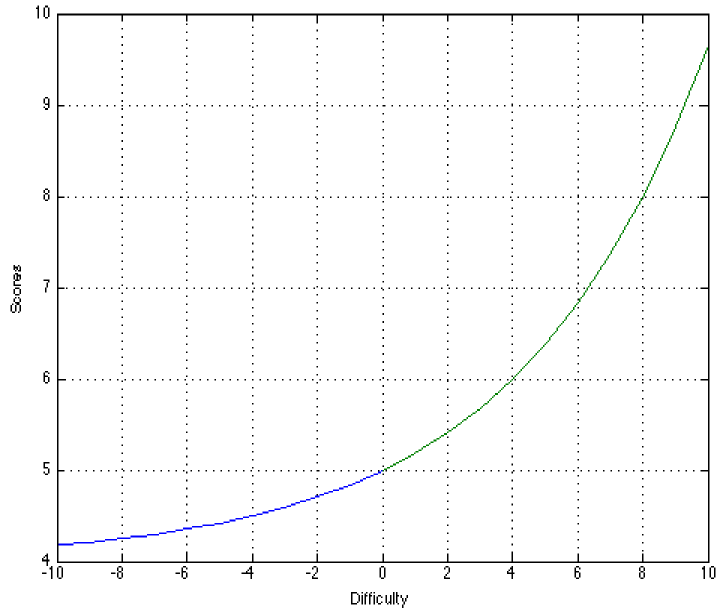

Figure 5.12: Non-Linear Model to Calculate the Score, Where the Score Range From 4.2 To 9.8.

5.5 SUMMARY OF RESULTS

This study was commissioned to estimate the ability of the student and to recommend ways of evaluating student performance. The research draws attention to the fact that IRT fails to estimate the ability when students answer all the questions correctly or when the student answers all the questions wrongly. While, the research suggested two methods which are one-dimensional random walk and hidden Markov model and it evaluates these models and concludes that it would be a satisfying solution to deal with the disadvantage of IRT. The models could estimate the ability in case of student answers all questions correctly or wrongly. Also, the models significantly reduced the information they need to estimate the ability compared with IRT.

6. CONCLUSION

In this thesis, we discussed performance of item response theory in computer adaptive testing. We explained how computer adaptive testing algorithm works and we addressed the benefits and limitations of item response theory. Firstly, we discussed the structure of item response theory and explain the three-parameter model by simulating the relationships between the ability level of the student, the difficulty level of the question and the probability of answering a question correctly.

Secondly, we suggested random walk theory to estimate the ability. We discussed the simplicity of the theory and its applications and we proposed two linear models and three nonlinear models of one-dimensional random walk. We simulated these models to illustrate the relationship between the difficulty and the ability. We find that the fifth ability estimation model (Figure 5.6) is more suitable, the ability grows exponentially when the difficulty increases. Also, we proposed hidden Markov model to estimate the ability of the student by simulating the Baum Welch algorithm. Moreover, we used HMM to predict the difficulty of the next question. In addition, we presented five different models to calculate the score for each question by simulating the relationship between the difficulty of the question and the grading points. We find that the fifth scoring model (Figure 5.12) is more convenient, the scores in the model are in the positive interval, which means the student gets positive scoring points whenever he answers any question correctly.

Thirdly, one-dimensional random walk models and HMM model need less information to estimate the ability which means they are straightforward and better models to improve computer adaptive testing compared with the complicated IRT.

Finally, many different tests, and experiments have been left for the future due to lack of time. The future work will be devoted to the development of an application to assessment students' performance. In the application, we would experiment with real data to test the methods that we proposed above to estimate the ability. Also, we would use the same methods as algorithm for selecting the optimal questions. This provides a good starting point to build an actual CAT application that has questions database and these questions will be addressed according to level of the difficulty.

Moreover, the difficulty level will be measured by statistical methods given a real set of responses from the students. Also, we consider increasing the difficulty of question by adding time factor, we could observe the response time and predict the response time for the next question. In addition, we would like to make a comparison between random walk models and HMM and based on the performance speed and the accuracy of estimating the ability and the difficulty of the questions.

REFERENCES

[1] S. Feinberg and W. van der Linden, "Statistical Models for Test Equating, Scaling, and Linking." New York, NY: *Springer New York*, 2011.

[2] L. Vinet and A. Zhedanov, "A 'missing' family of classical orthogonal polynomials," *J. Phys. A Math. Theor.*, Feb. 2011, vol. 44, no. 8, p. 85201.

[3] W. J. van der Linden and G. A. W. Glas, Eds., "Computerized Adaptive Testing: Theory and Practice". Dordrecht: *Springer Netherlands*, 2000.

[4] H. Wang, C. Ma, and N. Chen, "A brief review on Item Response Theory models-based parameter estimation methods," in *2010 5th International Conference on Computer Science & Education*, 2010, no. 1, pp. 19–22.

[5] J. M. Linacre, "Computer-adaptive testing: A methodology whose time has come," in *Development of Computerised Middle School Achievement Tests*, 2000, vol. 69, U. Kang, E. Jean, and J. M. Linacre, Eds. Chicago, IL. USA: MESA.

[6] D. Weiss, "Computerized adaptive testing for effective and efficient measurement in counseling and education," *Meas. Eval. Couns. Dev.*, 2004, vol. 37, no. 2, pp. 70–84.

[7] M. Antal, L. Erős, and A. Imre, "Computerized adaptive testing: implementation issues," *Financ. Anal. J* , Nov 2010, vol. 2, pp. 168–183,.

[8] R. Watson, L. A. van der Ark, L.-C. Lin, R. Fieo, I. J. Deary, and R. R. Meijer, "Item response theory: how Mokken scaling can be used in clinical practice.," *J. Clin. Nurs.*, Oct. 2012, vol. 21, no. 19–20, pp. 2736–46,.

[9] F. B. Baker, "The Basics of Item Response Theory". 2001.

[10] J. Rudnick and G. Gaspari, "Elements of the Random Walk", *Cambridge: Cambridge University Press*, 2004, vol. 119, no. 1–2.

[11] E. F. Fama, "Random Walks in Stock Market Prices," *Financ. Anal. J.*, vol. 1995, 51, no. 1, pp. 75–80.

[12] J. Masoliver, M. Montero, and G. H. Weiss, “Continuous-time random-walk model for financial distributions,” *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.*, 2003, vol. 67, no. 2, p. 10.

[13] T. GOTO and T. NAKAI, “J0260201 A discrete model of biased random walk based on bacterial chemo-taxis,” *Proc. Mech. Eng. Congr. Japan*, vol. 2015, p. _J0260201--_J0260201-.

[14] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh, “WTF: The Who to Follow Service at Twitter,” *WWW 2013 Proc. 22nd Int. Conf. World wide web*, 2013., pp. 505–514.

[15] L. Grady, “Random walks for image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, vol. 28, no. 11, pp. 1768–1783.

[16] G. Han and H. Sethu, “Waddling random walk: Fast and accurate mining of motif statistics in large graphs,” *Proc. - IEEE Int. Conf. Data Mining, ICDM*, 2017, pp. 181–190,.

[17] S. Russell and P. “Norvig, Artificial Intelligence A Modern Approach.” 2013.

[18] S. Ross, “Introduction to Probability Models.” 2014.

[19] J. Pan, M. T. Durand, B. J. Vander Jagt, and D. Liu, “Application of a Markov Chain Monte Carlo algorithm for snow water equivalent retrieval from passive microwave measurements,” *Remote Sens. Environ.*, 2017, vol. 192, pp. 150–165.

[20] B. Bellalta, “Throughput analysis in networks of WLANs,” *IEEE Commun. Lett.*, Nov. 2016, vol. 21, no. 3, pp. 592–595,.

[21] M. Shen, M. Wei, L. Zhu, and M. Wang, “Classification of Encrypted Traffic with Second-Order Markov Chains and Application Attribute Bigrams,” *IEEE Trans. Inf. Forensics Secur.*, 2017, vol. 12, no. 8, pp. 1830–1843.

[22] J. Zhu, T. Zhao, and Y. Zhang, “A Markov chain based method for probabilistic load flow with wind power,” in *2017 IEEE Power & Energy Society General Meeting*, 2017, no.

2015, pp. 1–5.

[23] L. E. Baum and T. Petrie, “Statistical Inference for Probabilistic Functions of Finite State Markov Chains,” *Ann. Math. Stat.*, 1966, vol. 37, no. 6, pp. 1554–1563.

[24] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, 1989, vol. 77, no. 2, pp. 257–286,.

[25] M. Kim, Y. Kim, J. Yoo, J. Wang, and H. Kim, “Regularized Speaker Adaptation of KL-HMM for Dysarthric Speech Recognition,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, Sep. 2017, vol. 25, no. 9, pp. 1581–1591.

[26] S. S. VijayaRajSolomon, V. Parthasarathy, and N. Thangavelu, “Exploiting acoustic similarities between Tamil and Indian English in the development of an HMM-based bilingual synthesiser,” *IET Signal Process.*, May 2017, vol. 11, no. 3, pp. 332–340.

[27] R. Raman, P. K. Sa, B. Majhi, and S. Bakshi, “Direction Estimation for Pedestrian Monitoring System in Smart Cities: An HMM Based Approach,” *IEEE Access*, 2016, vol. 4, pp. 5788–5808.

[28] L. Sun, J. Jiang, G. Zhang, and X. Wei, “A Discrete HMM-Based Feature Sequence Model Approach for Star Identification,” *IEEE Sens. J.*, Feb. 2016, vol. 16, no. 4, pp. 931–940.

[29] W. Lim *et al.*, “Dynamic Bandwidth Allocation for OFDMA-PONs Using Hidden Markov Model,” *IEEE Access*, 2017, vol. 5, pp. 21016–21019.

[30] M. Soruri, J. Sadri, and S. H. Zahiri, “Gene clustering with hidden Markov model optimized by PSO algorithm,” *Pattern Anal. Appl.*, Mar. 2018.

[31] I. Saif, Y. Kasmi, K. Allali, and M. M. Ennaji, “Prediction of DNA methylation in the promoter of gene suppressor tumor,” *Gene*, Apr. 2018, vol. 651, no. December 2018, pp. 166–173.

[32] R. L. Putland, L. Ranjard, R. Constantine, and C. A. Radford, “A hidden Markov model approach to indicate Bryde’s whale acoustics,” *Ecol. Indic.*, Jan. 2018, vol. 84, no.

September 2017, pp. 479–487.

[33] H. Zhang, W. Zhang, A. Palazoglu, and W. Sun, "Prediction of ozone levels using a Hidden Markov Model (HMM) with Gamma distribution," *Atmos. Environ.*, Dec. 2012, vol. 62, pp. 64–73.

[34] Weisstein, Eric W. Random Walk--1-Dimensional, From MathWorld--A Wolfram Web Available: http://mathworld.wolfram.com/RandomWalk1-Dimensional.html

[35] Undergraduate machine learning 9: Hidden Markov models HMM – University of British Columbia. Available: http://www.cs.ubc.ca/~nando/340-2012/lectures.php