ENHANCING PREDICTION ACCURACY OF A MULTI-CRITERIA RECOMMENDER SYSTEM USING ADAPTIVE GENETIC ALGORITHM

A Thesis Presented to the Department of

Computer Science

African University of Science and Technology

In Partial Fulfilment of the Requirements for the Degree of

Master of Science

By

Abdulsalam Ometere Latifat

Abuja, Nigeria

December, 2017

CERTIFICATION

This is to certify that the thesis titled “Enhancing Prediction Accuracy of a Multi-criteria Recommender System Using Adaptive Genetic Algorithm” submitted to the school of postgraduate studies, African University of Science and Technology (AUST), Abuja, Nigeria for the award of the Master's degree, is a record of original research carried out by Abdulsalam Ometere Latifat in the Department of Computer Science.

.

ENHANCING PREDICTION ACCURACY OF A MULTI-CRITERIA RECOMMENDER SYSTEM USING ADAPTIVE GENETIC ALGORITHM

By

Abdulsalam Ometere Latifat

A THESIS APPROVED BY THE COMPUTER SCIENCE DEPARTMENT

RECOMMENDED: ______________________________

Supervisor, Prof. Mohammed Hamada

____

Co-supervisor

____

Head, Department of Computer Science

APPROVED: ______________________________

Chief Academic Officer

____

December 9th, 2017

ABSTRACT

Recommender systems are powerful intelligent systems considered to be the solution to the problems of information overload. They provide users with personalized lists of recommended items, using some machine learning techniques. Traditionally, existing recommender systems have used single rating techniques to estimate users' opinions on items. Because user preferences might depend on the attributes of several items, the efficiency of traditional single-rating recommender systems is considered to be limited, since they cannot account for various items' attributes. A multi-criteria recommendation is a new technique that uses ratings of various items' attributes to make more efficient predictions. Nevertheless, despite the proven accuracy improvements of multi-criteria recommendation techniques, research needs to be done continuously to establish an efficient way to model criteria ratings. This project, therefore, propose to use an adaptive genetic algorithm to model multi-criteria recommendation problems, using an aggregation function approach. The anticipated empirical results of the project show that the proposed approach provides more accurate predictions than traditional recommendation approaches.

Keywords: recommender systems, multi-criteria recommendation techniques, prediction accuracy, adaptive genetic algorithm.

ACKNOWLEDGEMENTS

Firstly, I thank Almighty Allah, to whom all praise is due; the most beneficent, the most merciful and the Lord of the Worlds, for sparing my life and enabling me to complete this Master of Science degree program in good health.

I thank the African Capacity Building Foundation (ACBF) for considering me worthy of a scholarship for the Master of Science degree program, out of the many qualified candidates who merited the award. Your faith in me has yielded and will continue to yield the expected dividends.

I would also like to express my deepest appreciation to my supervisor, Professor Mohamed Hamada, whose guidance, encouragement and persistence, made this thesis possible. He is continually and convincingly a ‘spirit of adventure’ in regard to research; and always found the time to provide me with detailed explanations, suggestions and directions. I also offer my sincere appreciation to Mr. Mohammed Hassan (PhD), who took time out to help me despite his busy schedules. Thank you so much, Sir.

My profound gratitude goes to all the faculty members of the Computer Science and Engineering Department, especially my H.O.D. Professor Amos David, as well as Professor Lehel Csato, Professor Ben Abdallah and my supervisor, Professor Mohamed Hamada. I greatly appreciate your constructive criticism and the knowledge you imparted to me.

My appreciation and prayers for my parents will never cease: Alhaji and Hajiya Abdullahi Abdulsalam, who brought me up in the best way possible and supported me with all they possessed.

Also, to my dearest relations, Zainab, Yunusa, Awal, Musa, Nafisat, Mariam, my cutest nephew, and niece Adaeiza and Onyize, my brother-in-law Mr. Oba, and my uncles, Uncle Idris and Prof. Dauda, who have always been there for me, you are all the best. I greatly value your prayers, advice, and mental support before and during the course of my program. I pray that Allah reward you all abundantly.

I further extend my appreciation to all my friends - Khadijah, Habibah, Mulikat, Rukayya, Nkiruka, Folajomi, Ibrahim and Joseph, and also my course mates for their help, words of encouragement and prayers throughout the period of the program. You all made my stay one I would always remember.

Finally, to the entire management and staff of the African University of Science and Technology at Galadimawa in Abuja, Nigeria, I express my sincere gratitude for providing a learning-and-research environment within which my colleagues and I were able to conduct our Master of Science programs.

DEDICATION

This thesis is dedicated to Allah (SWT) and to my beloved parents, Alhaji Abdullah Abdulsalam and Hajiya Rabiat Ahmed Abdulsalam.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANFIS	Adaptive Neuro-Fuzzy Inference System
AP	Average Precision
ASVD	Asymmetric Singular Value Decomposition
AUC of ROC	Area under the Curve of ROC
DEA	Data Envelopment Analysis
DCG	Discounted Cumulative Gain
DMU	Decision-Making Unit
FCP	Fraction of Concordant Pairs
GA	Genetic Algorithm
HOSVD	Higher-Order Singular-Value Decomposition
IMCCF	Item-based Multi-Criteria Collaborative Filtering
MAE	Mean Absolute Error
MAP	Mean Average Precision
MAPE	Mean Absolute Percentage Error
MCDM	Multi-Criteria Decision-Making
MCRS-AGA	Multi-Criteria Recommender System based Adaptive Genetic Algorithm
MRR	Mean Reciprocal Ranking
MSE	Mean Squared Error
NDCG	Normalized Discounted Cumulative Gain
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic
RS	Recommender System
SVD	Singular Value Decomposition
SVM	Support Vector Machine

CHAPTER ONE: INTRODUCTION

1.1 Introduction

Intelligent systems are systems that require knowledge organisation to interpret, test and analyse acquired information. Intelligent systems are required in most of our day-to-day activities, such as e-commerce, online-booking, social media, e-shopping and other information-rich environments. Recommender systems interact with users in a personalized way, obtain information about a user's tastes or preferences and use this knowledge to make suggestions and provide assistance in situations where users have to make a decision between a wide range of possible options. In this chapter we endeavour to explain the recommender system and its techniques, introduce multi-criteria recommender systems and also a genetic algorithm. Statement of the problem, aims and objectives, significance and scope of this study will also be introduced in this chapter.

1.2 Background of the study

The recommender system was identified as a free research area in the mid-1990s, when researchers started concentrating on recommendation glitches that obviously depend on rating structure (Adomavicius & Tuzhilin, 2005). Recommender systems (RSs) are techniques and software tools for interacting with large and complex information spaces in order to prioritize and make suggestions on items, offers and objects likely to be of interest to a specific user (Ricci, Rokach & Shapira, 2015). These suggestions relay to several decision-making procedures, for example what item or object to buy, which movie to watch, what news to read online, what music to listen to, which airline to fly with or hotel to book (Ricci et al., 2015). Thus, the diversity in the feature of homogeneous products or services, related information and the choices available in the market place or in diverse application domains such as e-commerce, e-learning, e-government and e-tourism has made the recommender system broadly utilized (Shambour, Hourani & Fraihat, 2016).

Accuracy in the recommender system is a valuable factor in determining how it can effectively acquire and process information. This has made the evaluation of the recommender system a critical and challenging task. One major way of performing an evaluation of the recommender system is through accuracy (Sohrabi, Toloo, Moeini & Nalchigar, 2015). This project is aimed at developing an adaptive genetic algorithm to enhance prediction accuracy and obtain high correlation between predicted and actual values of a multi-criteria recommender system.

1.2.1 Recommender system techniques

The recommender system can vary in terms of the knowledge base, addressed domain, algorithm or technique used during development. According to Burke (2002), the recommender can be classified into six different approaches:

Content-based: In this approach the system learns from the user's previous likes and interests, then recommends matching items to the user based on that knowledge. The content-based approach places reliance on the item features and thus a learning method is employed to determine the type of user profile that will be derived by the content-based recommender. The similarity of the items is dependent on the features associated with the items (Ricci et al., 2015).

Collaborative filtering: This approach is the most prominent, developed and widely implemented technique (Burke, 2002). It generates recommendation of items to the active user based on previous items liked by other users with similar preferences. Collaborative filtering is called people-to-people correlation because the similarity in preference of two users is dependent on the similarity in the rating history of the users (Ricci et al., 2015).

Demographic: The main aim of this approach is to categorize the user based on personal attributes and to recommend items based on a user's demographic profile (Ricci et al., 2015). It may not require a history of user ratings.

Knowledge-based: This system recommends items based on a specific field of knowledge of how useful an item is to the user and how certain item features meet the needs and preferences of the user. The similarity measure can be interpreted as the utility of the recommendation.

Community-based: This approach provides recommendations on items to the user based on the preferences of the user's friends. It is suggested that people tend to rely more on the recommendations of friends than those of anonymous individuals with similar preferences (Ricci et al., 2015).

Hybrid recommender systems: This type of recommender system is a combination of two or more recommendation techniques mentioned above (Adomavicius & Tuzhilin, 2005). A hybrid system combining two techniques tries to use the advantages of one to solve the drawbacks of the other.

1.2.2 Multi-criteria recommender system

Traditionally, most RSs obtain an overall or general preference of a particular item by the user. In other words, it recommends items based on a single criterion rating by the users as the input information to be used by the RS algorithm to evaluate user preference opinions. In most cases, a single criterion rating could produce recommendations that do not meet the needs of the user because users can express their opinions based on some specific features of the item.

In contrast, multi-criteria RSs give users the opportunity to specify their preferences for an item based on multiple attributes (Ricci et al., 2015). Multi-criteria ratings provide additional information about preferences of the user regarding several important aspects or components of an item (Adomavicius & Kwon, 2007). The additional information on each user's preferences will lead to more accurate recommendations and will improve the quality of recommendations.

In recent years, multi-criteria ratings have been adopted by several recommender systems, instead of the traditional single criterion ratings (Ricci et al., 2015). The aim of multi-criteria recommender systems is to take a step towards analysing and understanding users' interests and choices in a more efficient and exquisite manner and providing the users with optimal solutions.

1.2.3 Genetic algorithm

In the 1950s and the 1960s, several computer scientists independently studied evolutionary systems with the idea that evolution could be used as an optimization tool for engineering problems. The idea was to evolve a population of candidate solutions to a given problem using operators inspired by natural genetic variation and natural selection (Mitchell, 2004).

A genetic algorithm is an evolutionary stochastic search method applied to optimization and learning. It is also a search algorithm based on the hypothesis for moving from one population of "chromosomes" to a new population by using a kind of natural selection (survival of the fittest) and natural genetics, which can be used to solve an optimization problem. Additionally, a genetic algorithm is an evolutionary approach to solving optimization problems such as sequencing, travelling, salesman problems and scheduling (Schmitt, 2001). In a genetic algorithm there are some important components that should be kept in consideration, such as:

Representation: The way individuals are defined, either in bit string, binary or real numbers.

Fitness function: Concerned with the measure of performance, which can be either minimized or maximized?

Population: This holds the representation of possible solutions.

Parent selection mechanism: Helps to distinguish individuals based on their quality, i.e., allowing the better individual to become parent of the next generation.

Variation operators: These operators create new individuals from old ones. They are divided into *crossover* (single point or two points), which is done on selected individuals to mix the generic information to get new individuals or offspring and *mutation* (flipping).

Selection mechanism: Is often called replacement and it is based on survival of the fittest.

Since genetic algorithm is stochastic and at most times guarantees no optimum solution, a suitable termination condition is required, which could be when the fitness evaluation reaches a given limit.

1.3 Statement of the problem

The majority of prevailing RSs uses an overall estimation of a user rating of an item or single criterion rating techniques to evaluate users' opinions on experienced items. Since the suitability of the recommended item for a particular user may depend on several important aspects or attributes in the decision making of the user, the efficiency of the traditional single criterion rating can be deliberated to be limited and inaccurate, because it cannot justify for the various items' attributes.

For this purpose, a multi-criteria recommendation which implements users’ ratings on multiple or various attributes of an item using aggregate function-based approach is proposed. The proposed technique acquires an appropriate learning relationship using an adaptive genetic algorithm to achieve a more accurate and efficient prediction.

1.4 Aim and objectives of the study

The aim of this project is to use an adaptive genetic algorithm to model a multi-criteria recommendation problem using an aggregation function-based approach to achieve a more accurate and efficient prediction.

The specific objectives were:

- To formulate an adaptive genetic algorithm model.
- To use an adaptive genetic algorithm to model multi-criteria recommendation problems.
- To develop a system that will be proficient enough to recommend the most appropriate item to a user.
- To compare the predictive performance of the multi-criteria recommender technique using an adaptive genetic algorithm with the traditional recommender approach.

1.5 Significance of the study

- Web users, application domains such as e-commerce, e-learning, e-government, social networks and e-tourism are the main benefactors of this study due to the fact that the system further creates an easier, faster and more efficient decision-making strategy.
- A user rating of an item with multiple attributes based on his or her personal interest can efficiently improve the prediction accuracy of the recommendation to other users.

1.6 Scope of the study

The scope of the study is on RSs, with emphasis on multi-criteria RSs that will generate a proficient, accurate prediction for users. This study entails the development of a sophisticated system capable enough to recommend the most appropriate suggestion or item to users based on their preferences.

1.7 Expected results

The project aims to provide the predictive performance of the proposed technique and compare it with that of existing methods. These performances include a decrease in prediction errors, increase in ranking accuracy and high correlation between predicted and actual values.

1.8 Thesis structure

The rest of this thesis is organised as follows: Chapter 2 presents an overview of the RS and multi-criteria RS, discusses the adaptive genetic algorithm and its component, and reviews related studies. Chapter 3 describes the methodologies and architecture of the study. Chapter 4 presents the detailed implementation of the system. It also discusses the results obtained. Chapter 5 wraps up by discussing the summary, conclusion and recommendation.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

The Recommender Systems (RSs) and its application, as well as the overview of collaborative filtering, will be discussed. We will also present the genetic algorithm and its component multi-criteria RS and provide a review of existing relevant literature. The literature review was carried out in order to identify what has been done before on the subject matter as regards this thesis, highlight weaknesses and suggest how the proposed system to be developed intends to solve these identified weaknesses.

2.2 Overview of Recommender Systems and their applications

RSs are technology-based systems that primarily use relevant ratings data given by a user to make personalized recommendations to active users. RSs make use of opinions and actions of other users with similar preferences to build recommendations, thereby helping users avoid information overload. These systems contain ubiquitous features in most e-commerce sites such as Netflix.com, Ebay.com, Amazon.com, Instagram, Facebook and Last.fm. RSs are information-processing systems that gather numerous kinds of data (the items to suggest and the users who will receive the recommendations) and knowledge sources in order to generate recommendations for users. These data used by the RSs refer to three kinds of objects: items, users and transactions (Ricci et al., 2015).

A RS is a technique by which an algorithm is used to detect or predict what a user is buying. One would always prefer that recommendations made are similar to the user’s taste or items already bought in the past. Recommendation engines help to do just that. The first factor to consider in developing any recommendation system is its application domain, which has a major effect on the algorithmic approach taken (Ricci et al., 2015). RSs help users to find interesting items in a given domain (such as movies, books, applications, music, hotels and restaurants) by recommending items that match their preferences. Important application domains have been mentioned, but below is a centralized list of the application of RSs:

E-commerce or product recommendation: Ebay.com, Ali-express and Amazon, which are big online stores that sell a large variety of products, use recommendation techniques to display a list of recommended items a user might be interested in, by using the information from users' past actions, or decisions made by similar users. In addition, users can rate back their purchases using a five-star rating scale, thereby improving the accuracy of the recommendations.

Social networks: Facebook, LinkedIn, Twitter and Instagram use recommender techniques to identify and recommend a list of people or to predict "people the user may know" in order to increase the number of social connections on the site. These recommended people can be based on mutual friends or information, such as school attended, location and place of work, a user might have provided on his/her profile. They are also platforms for advertisement.

Entertainment domain: Life Trak and Last.fm are music platforms that create an environment for users to listen to music and provide feedbacks or ratings. These platforms provide recommendations to users based on feedbacks and ratings received. Yahoo! Movies, IMDb and Movie Lens are movie platforms that provide users with the opportunity to watch movies, stream movies, rate the movies and recommend to other users' movies they might like. Netflix is a platform that provides video streaming, DVD-rental services and television shows on subscription basis, and it provides an avenue for users to rate movies and television shows on a five-point scale.

News articles: These platforms, such as Google News and Group Lens, attempt to identify articles of interest to readers based on the articles they have read in the past. A similar principle is also applied in recommending blogs from among millions of blogs available, videos on YouTube or other sites where content is provided on regular basis.

Automated RSs have revolutionized the marketing strategy and delivery of commerce and its content by providing personalized recommendations and predictions over a wide variety of large and complex product offerings (Konstan & Riedl, 2012).

2.3 Ratings

Ratings can be said to be the degree of preference or a form of feedback of a user in which numerical values are selected from a specific evaluation system that specifies their likes and dislikes of various items. The design of a RS is highly influenced by the system used for tracking ratings, and these ratings are specified based on the level of like and dislike of a user. These ratings may be collected:

Explicitly: This is the case where the user is asked to provide an opinion or preference about an item on a rating scale and the accuracy of the recommendation is dependent on the quantity of ratings provided by the user.

Implicitly: In this ratings collection, the system aims to infer the user's preferences based on the user's actions or activities.

2.4 Types of rating

Ratings can take different forms:

Numerical ratings: This type of rating is provided in the book *recommender associated with Amazon* and is also used by Netflix. A user is expected to rate on a five-star scale and alongside each possible rating is indicated a semantic interpretation of the user's level of preference.

Ordinal ratings: Ordered, categorical values such as "Strongly Disagree, Disagree, Neutral, Agree and Strongly Agree". Ordinal ratings are usually done via a questionnaire, where a user is asked to select the term that best indicates his or her opinion regarding an item.

Binary ratings: These ratings model choices in which the user is simply asked to represent a dislike for certain items and nothing else, or rate whether an item is good or bad, i.e., 1 or 0.

Unary ratings: They are a special case of rating common in the case of implicit feedback data sets. They create a mechanism which specifies that a user has observed or purchased an item or otherwise rated the item positively, but no mechanism to specify when there is an absence of rating or dislike, which shows there is no information relating the user to the item.

2.5 Measure of accuracy

A basic assumption in a RS is that a system that provides more accurate predictions is going to be preferred by the user, leading to considerable research work on how to provide or generate algorithms that will yield better prediction accuracy. RSs are prone to two most important problems: *rating prediction* (predicting the rating that a user *u* will give to an unrated item *i*) and *top-N* (a rank-based measure that focuses on only the ranking of top-N items rather than all items) recommendation problems (Ricci et al., 2015**)**. Accuracy is commonly used to evaluate the performance of the recommendation method, and these evaluations can be performed by measuring the correctness of rating predictions or by measuring the accuracy of ranking the recommended items. Recommendation accuracy metrics can be broadly classified in three (3) classes (Herlocker, Konstan, Terveen & Riedl, 2004), namely:

Predictive accuracy metrics: Predictive accuracy metrics measure how close the recommender system's predictions are to the exact ratings. Predictive accuracy metrics are very important for evaluating non-binary ratings and are appropriate for situations in which an accurate prediction of the ratings of all items is of high importance. The most important types of this class are *C* (Schröder, Thiele & Lehner, 2011). MSE and RMSE place more emphasis on larger errors, compared to MAE metrics. Predictive accuracy error metrics are frequently used in RSs because they are easy to understand and compute.

Classification accuracy metrics: Classification accuracy metrics measure the frequency of exactness or correctness of a recommendation**.** They measure the amount of correct and incorrect decisions made by the RS as relevant or irrelevant items. This helps to make classification metrics useful and appropriate to user tasks by finding good items, especially when users have true binary preferences. Classification metrics also attempt to measure the completeness of an algorithm and are particularly suitable for application in ecommerce (Schröder et al., 2011). Classification metrics are of different types, including: *Precision*, *Recall*, *F1-measure*, *Average Precision* (AP), *Mean Average Precision* (MAP), *Receiver Operating Characteristic* (ROC) etc. The *F1-measure* is the standard harmonic mean of precision and recall and the MAP is the arithmetic mean of AP.

Rank accuracy metrics: Rank accuracy metrics measure the ability of a RS to estimate the correct order of items that match how the user would have ordered the same items. Rank accuracy metrics can also be said to be an extension of precision and recall taking the position of correct items in a ranked list into account. They are more appropriate for evaluating algorithms that will be used to provide ranked recommendation lists to the user in situations where the user's choices are not binary (Herlocker et al., 2004). Ranking accuracy metrics are of different types, including: *Discounted Cumulative Gain* (DCG), *Normalized Discounted Cumulative Gain* (NDCG), *Mean Reciprocal Ranking* (MRR), *Fraction of Concordant Pairs* (FCP), *Area under the Curve of ROC* (AUC of ROC), etc.

2.6 Overview of collaborative filtering

Collaborative filtering is the most prominent approach to generating recommendations. It is based on the nature of human "use what is trendy among my peers or use the wisdom of the crowd" to recommend items. It focuses on the relationship between users and items. Collaborative filtering tends to work well with complex objects such as music and movies and it is completely independent of any machine-readable representation of objects being recommended. Collaborative filtering is used by large commercial e-commerce sites and is also applicable in many domains ranging from movies, books, DVDs, music, restaurants, hotels, webpages and articles to grocery products, and so forth. The fundamental assumption of collaborative filtering is that customers or users with similar tastes in the past will have similar tastes in the future. For example, if users A and B rate *n* items similarly or have similar preferences in terms of buying, listening or watching an item, they will tend to rate or act on other items similarly.

2.6.1 Types of collaborative filtering

RS engines have been deployed by various companies and depend heavily on various types of algorithm that vary significantly across the system, ranging from the classical collaborative filtering deployed by Amazon and Net Perceptions to the statistical relational learning methods deployed by Clever Set (Adomaviciuszan & Tuzhilin, 2014). Algorithmically, collaborative filtering techniques can be broadly classified in two categories, as discussed below.

2.6.1.1 Memory-based techniques

The memory-based approach is also known as neighbourhood-based collaborative filtering. They are techniques that make calculations based on users' stored data or previous activities. They usually represent heuristics that calculate recommendations on the fly, based on the activities of previous user or stored data (Adomavicius & Kwon, 2007). The rating matrix of model-based approaches is directly used to find neighbours or make predictions and they do not scale for most real-world scenarios.

Memory-based approaches usually make use of similarity metrics to obtain the distance between two items or two users, based on each of their items, and its results are always updated. There are primarily two ways to achieve a memory-based approach: user-based and item-based techniques (Isinkaye, Folajimi & Ojokoh, 2015). The user-based collaborative filtering technique utilizes the rating of similar users by comparing their ratings on the same item, for example if Bill and Job have rated a movie in the past in a similar way, then one can use Bill's observed rating on a movie, *The Boss Baby,* to predict Job's rating on the same movie. The item-based collaborative filtering technique computes predictions using the similarity between items, i.e. ratings given to similar items. For example, Bill's rating of adventure movies such as *The Jungle book* and *Moana* can be used to predict his rating of *Jumanji*.

2.6.1.2 Model-based techniques

Model-based techniques have been reported to achieve higher accuracy, in that they are based on offline pre-processing or use machine learning from previous stored data. Model-based techniques use previous user activities or stored data to first learn a predictive model (the model is trained with past activities) either using a machine-learning method or some statistical method, which the system uses to make recommendations (Adomavicius & Kwon, 2015). In these approaches at run time, only the learned model is used to make predictions and the models are updated or re-trained periodically. Some model-based approaches include Bayesian classifiers, neural networks, fuzzy patterns, genetic algorithms, latent features and matrix factorizations (such as singular value decomposition (SVD). In this project, since the aim is to improve accuracy, for finding our single-criteria predicted ratings we will focus on model-based techniques, especially on the matrix factorization using the singular value decomposition approach.

The main aim of these two techniques (memory-based and model-based approaches) is to get the most accurate predictions based on users' preferences, and the accuracy of these predictions can be evaluated through classical information retrieval measures such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Most researchers use these measures in order to improve the technologies and methods of a RS (Bobadilla, Ortega, Hernando & Gutiérrez, 2013).

Singular Value Decomposition

Recently, Singular Value Decomposition (SVD) models have gained popularity mainly because of their attractive accuracy and scalability; and they also serve as a powerful tool for dimensionality. Matrix factorization models map both users and items to a joint latent factor space of dimensionality f, the latent space tries to explain ratings by identifying both items and users on factors automatically inferred from the feedback of the user (Ricci, Rokach, & Shapira, 2015). A typical SVD model associates each item *i* with an item-feature vector $L_i \in \mathbb{R}^f$ and each user *u* with a user-feature vector $M_u \in \mathbb{R}^f$, where f denotes the feature number (Koren, Ave, Park, Management & Applications, 2008). The prediction of an item to a user is done by:

$$R'_{ui} = b_{ui} + L_i{}^T M_u \qquad (1)$$

Where $b_{ui} = b_u + b_i + \mu$ (2)

Thus, $R'_{ui} = b_u + b_i + \mu + L_i{}^T M_u$ (3)

Where R'_{ui} is the predicted rating given by user *u* to item *i*, b_{ui} is the baseline estimate of an unknown rating R'_{ui}, μ is the global average overall rating, b_u and b_i indicate the observed deviation of user *u* and item *i* respectively. Thus, learning this model means finding feature vector. To learn the model parameter (such as b_u, b_i, and the feature vectors M_u and L_i) the system minimizes the regularized squared error on the set of known ratings R_{ui} (Bokde, Girase & Mukhopadhyay, 2014). The formula below shows the minimization:

$$\min_{b^*,L^*,M^*} \sum_{(u,i)\in K} (R_{ui} - \mu - b_u - b_i - L_i{}^T M_u)^2 + \lambda(b_u^2 + b_i^2 + \| L_i\|^2 + \|M_u\|^2) \qquad (4)$$

Where constant λ controls the regularization (it is normally determined by cross validation), and K is the set of pairs (u, i) for the known ratings given by user u to item i.

The minimization is performed either by using stochastic gradient descent or alternating least squares technique (Ricci et al., 2015). Applying stochastic gradient descent for each rating R_{ui}, the predicted rating R'_{ui} is obtained and the prediction error $E_{ui} = R_{ui} - R'_{ui}$ is measured (Hassan & Hamada, 2017). For a given training set R_{ui}, the parameters are modified by moving in the opposite direction of the gradient (Ricci et al., 2015), yielding:

$$M_u \leftarrow M_u + \gamma(E_{ui}L_i - \lambda M_u)$$
$$L_i \leftarrow L_i + \gamma(E_{ui}M_u - \lambda L_i)$$
$$b_u \leftarrow b_u + \gamma(E_{ui} - \lambda b_u)$$
$$b_i \leftarrow b_i + \gamma(E_{ui} - \lambda b_i)$$

The parameters γ and λ are assigned small positive real numbers.

Alternating least squares techniques rotate between fixing the L_i's to solve for the $M_u{}'s$ and fixing the $M_u{}'s$ to solve for the $L_i{}'s$, while the stochastic gradient descent loops through all known ratings in the training data (Ricci et al., 2015).

2.6.2 Application of collaborative filtering

Collaborative filtering can be applied in various domains. Examples of such include Group Lens, which uses the earliest automated collaborative filtering technique to filter users, and Netnews by building on news group browsers with a rating functionality for users. The ideas generated in Group Lens were extended to other product settings such as books (Book Lens) and movies (Movie Lens) (Candillier, Jack, Fessant & Meyer, 1997). In Google, new collaborative filtering algorithms are also applied to the collected rating to enable inferences to be made about personalized articles for specific users. Collaborative filtering was also applied in the Tapestry mailing and repository system to weave information, where users are provided with personalized mailing list filters instead of being pushed to subscribe.

In the aspect of social media where collaborative filtering is applied to recommend interesting or popular information as rated by the community, and content improves as the number of users increase or becomes more diverse, Reddit, YouTube and Last.fm are typical areas or domains of collaborative filtering-based media.

2.7 Genetic Algorithm

Genetic algorithm (GA) is a robust stochastic search method based on natural evaluation and natural selection ("survival of the fittest") and is often applied to optimization problems. GAs' main application is in solving combinatorial optimization and in development of parallel computers (Abdullah & Turabieh, 2008). GAs use information from a population of individuals, i.e., different individuals, to conduct research or solve optimization problems for better results. GA's are applied for those problems that either cannot be formulated in exact and accurate mathematical forms, which may contain noisy data, or for problems that are simply impossible to solve by applying traditional computational methods.

The idea behind a genetic algorithm is the same as other evolutionary algorithm techniques, where a population of individuals is, due to environmental pressure, subjected to the process of natural selection (survival of the fittest), which in turn results in the evolution of a population group that exhibits predominant fitness traits. A genetic algorithm uses information from a population of individuals, let us say *P(t)*, where *P* is the population and *t* the number of the individuals and each individual represents a potential solution to the problem to be solved. A fitness value is assigned from evaluating each of the individuals which depends on how close each individual is to solving the problem. As the iteration continues, the processes eventually approach a local maximum and minimum of the function. GA can achieve good results in cases where the functions have several multiple local maxima or minima.

GAs, unlike other evolutionary algorithms, are explicitly parallel, and thus cannot be trapped in a suboptimal local maximum or minimum of the target function since it can be carried out on different processors (Goodman, 2008). In genetic algorithms, the potential solution to the problem to be solved is usually referred to as the *chromosome*, which is usually encoded as bit string or binary string and is an abstract genetic representation. The binary encoding of chromosomes gives possible settings for a trait. The "genes" are either single bits or adjacent bits that encode a particular element of the potential solution: in other words, each chromosome is a collection of genes. A position that encodes some traits is called *Locus* and each possible value at a locus is called an a*llele* (Goodman, 2008). An allele in a bit string is either 0 or 1.

Genotype is a particular set of genes in a genome and *phenotype* is the physical expression of the genotype. The next step after encoding the chromosome, which can be done through either binary encoding, permutation encoding or real value encoding etc., is to apply different reproduction operations to the genotype and phenotype. Genetic operators (selection, crossover and mutation) work on genotype space. Finally, the process is completed provided that the stopping criteria have been fully satisfied. If not, it loops again.

2.7.1 Initial population

In GA, the first thing that is taken into consideration is population size and the method used to select the population group. The population size is important because it has an effect on the performance of the algorithm. It represents the number of individuals that will be initially formed and from which the next generation will be created. By selecting too small a population size, representation cannot be achieved, resulting in the lack of a sufficient avenue to explore the search space effectively, while too large a population will not yield a solution within a reasonable amount of time because it will take longer to yield a reasonable solution (Reeves, 1997). The method used to choose the population is usually a random selection

.

2.7.2 Fitness evaluation

A fitness function is a particular type of objective function that assigns a fitness value to a chromosome based on how close or how far the chromosome is from the optimum solution in a genetic algorithm, so that the particular chromosome may be ranked against all other chromosomes. The greater the fitness value of the chromosome, the better the solution it contains. The chromosomes with the optimal solution or most optimal solution is allowed to mix and breed to produce a new population that will hopefully be better than the previous. The fitness function is the most important aspect of the genetic algorithm because it decides on which chromosome is best fit or which individual is less fit. If a less fit individual is used by the genetic algorithm, it can lead to poor performance of the algorithm.

2.7.3 Selection

Selection in GA helps to provide a better individual to be parent of the next generation, and it is related to the fitness of the population. Selection is normally implemented using the roulette-wheel method and it uses a probability distribution for selection which helps to pair individuals in the order they are generated and is proportional to its fitness. The roulette-wheel method had some drawbacks, including a *high stochastic variability*, *difficulty finding a suitable fitness measure* for the members of the population and being *cumbersome*, so two alternatives were provided: *Ranking* or t*ournament selection*.

Ranking helped tune the selection pressure by selecting high-ranked individuals ranked on their fitness order. Although some information might be lost, ranking does not require re-scaling and makes selection easier and more efficient. Tournament selection compares a chosen set of individuals and picks the best of them for parenthood. It is arbitrarily stochastic in the sense that not all individuals will be available in the cycle at all times, but it has an advantage over all other forms in that it requires only a preference ordering between pairs of strings and can cope with situations where there is no formal objective function at all (Reeves, 1997).

2.7.4 Crossover

In GA, crossover is an operator that operates on two parent individuals in which some of the genes in one parent are replaced by corresponding genes of the other. It produces an offspring or two offspring, which can be the possible solution. A gene is a part of a chromosome that contains a part of a solution, in other words it determines the solution to a problem (Chakraborty, 2010). Crossover occurs at o*ne-point* or *two-point*. One-point crossover is a crossover in which one cross point is chosen at random from the given numbers and a new solution is produced by combining the pieces of the original parents, while in two-point crossover, two points are chosen at random from the given numbers and a new solution is produced by combining the pieces of the original parents (Goodman, 2008). For an example see Figure 2.1 below:

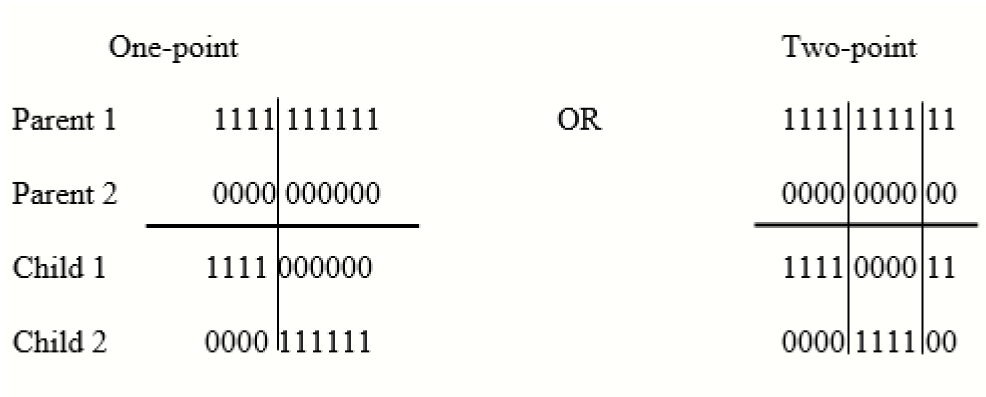


Figure 2.1: One Point and Two Point Crossover

2.7.5 Mutation

Mutation is a GA operator that operates on one parent individual in which an offspring or child is produced by flipping some of the bits in a chromosome (Mitchell, 2004). Mutation can occur at each bit position in a string, with each bit having the same probability of mutation. For example, the string 111100000111 might be mutated in its eighth position to yield 111100010111. Selection, crossover and mutation were carried out on a population of individuals in which a new set of individuals were produced that became the new population.

2.7.6 Termination

A genetic algorithm is stochastic in nature, which makes it difficult to guarantee an optimum solution; thus, the need for a termination condition arises; otherwise the algorithm may never stop. A suitable termination condition can be distinguished with two possible scenarios: if the problem has a known optimal fitness level and if it is coming from a known optimum of the given objective function, then reaching this level should be used as the stopping condition (Eiben & Smith, 2007). If the first condition is not satisfied, however, and the algorithm continues, then it requires that the condition be extended. Some commonly used options may include:

- When the stipulated CPU time expires.
- When the population diversity drops below the given threshold.
- When the total number of fitness evaluation reaches a given limit.

- For a number of generations, the fitness improvement remains under a threshold value (Eiben & Smith, 2007).

Encoding (From phenotype to genotype): This relates to placing a sequence of characters or providing a potential solution to the problem in a specialized format that will enable the computer to process. When a problem is to be solved with GA, encoding is the first step to be considered before attempting to solve the problem.

There are different ways of encoding that include binary encoding, permutation encoding, value encoding and tree encoding. Binary encoding is the most commonly used encoding approach because of its simplicity and large room for many possible candidate solutions. It is made up of a sequence of binary strings (such as 1s and 0s).

Decoding (From genotype to phenotype): this process is required after crossover and mutation. In GA, decoding is the conversion of individual chromosomes into binary, nominal, integer and real phenotype, which can be done using either Gray or base-2 encoding.

2.8 Cold start problem

One of the biggest problems of the traditional RSs is the cold-start and sparsity problem, which hinders recommendation generation due to a lack of sufficient information about items to be recommended to the user (Ricci et al., 2015). This in turn may lead to inaccuracy in prediction of items to users. When there is enough information on an item, i.e. if the item is rated based on multiple criteria, it will be easy to accurately recommend such an item to another user based either on a similar user's preferences or on the user's information.

2.9 Multi-criteria recommender systems

The traditional collaborative filtering RS makes use of single-rating or overall rating provided by a user for an item they have purchased in the past, as input to be evaluated and recommended or predicted to the active user. These recommendation types tend to be inaccurate and, in most applications, they do not meet users' needs; thus, there is a need for a systematic method that considers multiple ratings to evaluate the RSs. To address this problem, this project introduces a multi-criteria rating technique to evaluate the RS.

A single–rating system allow users to rate based on a single criterion and this information retrieved from the single-rating is limited or not enough, which can lead to inaccuracy. On the other hand, a multi-criteria system provides more information about users’ preferences by allowing users to rate products and services along various dimensions or criteria for an item. This additional information can potentially increase the recommendation accuracy or prediction accuracy and improve the quality of recommendation because it would be able to represent more complex tastes of each of the users. Thus, a multi-criteria system can provide robust tools for system designers to build more interesting systems (Nilashi, Ibrahim, Ithnin & Sarmin, 2015). The only difference between a multi-criteria rating and a single criterion rating is that the multi-criteria rating provides more information. A multi-criteria RS has two approaches:

2.9.1 Similarity-based approach

The similarity-based approach extends the traditional single-criteria collaborative filtering (in particular the memory-based collaborative filtering) by computing the similarity between two users using multi-criteria ratings (Adomavicius & Kwon, 2007). There exist two ways in which multi-criteria ratings can be used to compute similarity between two users: first by aggregating traditional similarities from individual criteria, which can use either a correlation-based method or a cosine-based method to calculate the similarity on the basis of individual criterion. Also, the overall similarity is computed using either an average similarity, worst-case (smallest) similarity or aggregate similarity. The second approach is by calculating similarity using multi-dimensional distance metrics, i.e., it changes only the similarity calculation component of a traditional recommendation algorithm. To achieve the second approach, there is a need to find the distance between two users’ ratings for the same item (either by Manhattan distance, Euclidean distance or Chebyshev distance), find the overall distance between two users (the average distance between their ratings for all their common items) and then transform the distance into similarity.

2.9.2 Aggregation function-based approach

The intuition behind this approach comes from the assumption that multi-criteria ratings represent user tastes for various dimensions or criteria of an item, i.e., it assumes the overall rating serves as an aggregate of multi-criteria ratings. This approach is not limited to any specific recommendation algorithm and infers that the overall rating has a relationship with the multi-criteria ratings. An aggregation function-based approach consists of three steps (Ricci et al., 2015). First, this approach predicts multi-criteria ratings using any recommendation technique by decomposing the k-dimensional multi-criteria rating space into k single-rating recommendation problems. Secondly, the aggregation function learns by using domain expertise, machine learning techniques (such as fuzzy logics and genetic algorithms) or statistical techniques. These different techniques are used for testing the accuracy or fitness of the predicted aggregation function.

Lastly, the approach predicts the overall rating by computing the unknown overall rating using the multi-criteria ratings estimated in the first step and the function estimated in the second step (Adomavicius & Kwon, 2015). The aggregation function approach has three (3) different scopes: Total (when a function is used to predict all unknown ratings); user-based (when a separate aggregation function is learned for each user and the user tends to benefit from having his own user-based aggregation function); and item-based (when a separate aggregation function is learned for each item). This approach tends to outperform the traditional single-rating collaborative filtering technique in terms of precision in a top-N metric (Ricci et al., 2015).

Multi-criteria rating systems have been successful in many applications and have been commonly adopted in many industries, i.e., online shopping malls such as Buy.com and Circuitcity.com, which provide multi-criteria ratings for consumer electronics (for example, performance, display size, cost and battery life). Yahoo! Movies launched a movie recommendation service that shows ratings by users of a movie for four specific criteria (for example, story, visuals, direction and acting). Restaurant guides such as Zagat's Guide provide three rating criteria for restaurants (for example, food, décor and services) (Ricci et al., 2015).

Other popular domains in which this system can also be applied include experiential products such as movies, books, travel and tourism, music and mobile banking. This shows that multi-criteria data can provide value for both online content providers and consumers or users. Multi-criteria rating systems may soon become an important component in different applications. In this project we propose a new approach on how to enhance the prediction accuracy of these multi-criteria RSs.

Our study focuses more on this approach because of its numerous advantages over the similarity based. An example is the ability of the aggregation function-based approach to work with any traditional collaborative filtering technique, while the similarity-based approach can work only with the traditional similarity-based collaborative filtering.

2.10 Review of related studies

In previous years, much research has been done in the area of RSs and their evaluation. A number of research studies have recently adopted a multi-criteria rating system for recommendations, because it can help boost recommendation accuracy and provide precise recommendations. Thus, the problems associated with multi-criteria recommendations have started to gain attention and are regarded as among the most important issues in evolving RSs. The most critical problems related to multi-criteria systems are prediction accuracy and multi-criteria optimization. Different research has been conducted to achieve an optimum solution. Some of the most important works are discussed below:

2.10.1 New recommendation techniques for multi-criteria rating systems

Adomavicius and Kwon (2015) proposed several new techniques for extending recommendation technologies to incorporate and leverage multi-criteria rating information in RSs. Two new recommendation approaches were proposed (similarity-based and aggregation function-based). Multiple variations of each proposed approach were discussed, and an empirical analysis of these approaches was performed using a real-world data set (Yahoo! Movies).

The criteria values (ratings) in the data set were initially presented using a 13-fold quantitative scale from A^+ to F, representing the highest and lowest preferences respectively.

The same manner was also used to change the rating representation to numerical form, i.e. 13 to 1 instead of A^+ to F. The cosine-based similarity technique was used to determine the similarity between two users, and numerous metrics for evaluating the performance were also proposed, including statistical accuracy metrics such as root mean squared error, as well as decision-support measures that determine how well the recommendation algorithm can predict high-relevance items (items that would be rated highly by the user). Precision-related metrics were mainly the focus in this paper; that is, precision-in-top-N, which represented the percentage of truly "high" overall ratings among those predicted to be N most relevant items for each user.

To illustrate the performance of proposed multi-criteria recommendation techniques, an empirical analysis of five approaches using the Yahoo! Movies data set was carried out. The five approaches were standard collaborative filtering, two similarity–based techniques (cos-min and Chebyshev) and two aggregation function-based techniques (total-reg and movie-reg 95). Standard user-based collaborative filtering was used as an integral part of every technique in order to minimize non-essential differences between them as much as possible and maximize any differences in performance there might be between standard collaborative filtering and multi-criteria RSs due to the newly introduced multi-criteria rating information.

The experimental results showed that multi-criteria ratings can be successfully leveraged to improve recommendation accuracy when compared with traditional single-rating recommendation techniques. Our proposed approach tends to extend these approaches with the aid of an adaptive genetic algorithm aimed at achieving a more precise, accurate prediction.

2.10.2 An item-based multi-criteria collaborative filtering algorithm for personalized recommender systems

Shambour, Hourani and Fraihat (2016) proposed an item-based multi-criteria collaborative filtering (IMCCF) algorithm that integrates the items' semantic information and multi-criteria ratings of items with lesser-known limitations of item-based collaborative filtering techniques. The proposed IMCCF algorithm took a raw matrix of user-item multi-criteria ratings as input, which consisted of multi-criteria ratings of *M* users on *N* items, and hierarchical tree-structured item taxonomy. The item taxonomy, given by domain experts, had a set of main item categories where items belonged to leaf nodes. The *Euclidean distance* similarity measure was used to calculate multi-criteria item-based collaborative filtering similarity values between the target item and its neighbour, based on each individual criterion. The standard vector-based cosine similarity was used to compute the value of the item-based semantic between two items. To validate the performance of the proposed IMCCF recommendation algorithm, Yahoo! Movies' data set was used.

Mean absolute error was used to measure prediction accuracy, and a coverage metric to evaluate the capability of a given recommendation algorithm to produce recommendations.

Two main experiments were performed to prove the improvement of the proposed IMCCF recommendation algorithm: (1) Evaluating the prediction accuracy and recommendation coverage of the IMCCF on the sparsity problem, and (2) evaluating the prediction accuracy and recommendation coverage of the IMCCF on the new item problem. Experimental results were compared to the results of two widely used item-based collaborative filtering algorithms from previous works. According to the experimental results, the proposed algorithm proved to be very effective in terms of dealing with the sparsity and new-item problems. They therefore produced recommendations that were more accurate in comparison to standard item-based collaborative filtering techniques. The proposed IMCCF algorithm enhanced the quality of produced recommendations by exploiting the added information obtained from the multi-criteria ratings of users and the semantic relationships among items to address sparsity and new-item limitations.

2.10.3 Multi-criteria collaborative filtering with high accuracy, using higher-order singular-value decomposition and a neuro-fuzzy system

Nilashi, Ibrahim & Ithnin (2014) proposed a new model for multi-criteria collaborative filtering using an adaptive neuro-fuzzy inference system (ANFIS), combined with subtractive clustering and higher-order singular-value decomposition (HOSVD) aimed at improving the prediction accuracy and recommendation quality of a multi-criteria collaborative filtering recommender system. HOSVD was used for dimensionality reduction to improve the scalability problem, while ANFIS was used to extract fuzzy rules from the experimental data set, alleviating the sparsity problem in overall ratings and representing and reasoning users' behaviour on item features.

Several experiments were conducted on Yahoo! Movies' data set and a test data set was also created.

HOSVD was used to effectively reduce the noise of high-dimensional data and improve the scalability problem. By using HOSVD, all factors in the third-order tensor of user, item and criteria were considered together to reveal latent relationships between them. A silhouette co-efficient value was adopted as the standard measure for clustering quality and used to determine the best cluster formation. The result of applying the HOSVD method to the high dimensional data set was to help them have clusters of high quality, using cosine-based similarity. ANFIS provided a flexible structure to the defined problem suitable for generating stipulated input-output pairs, using a set of induced fuzzy IF-THEN rules with appropriate and varied MFs. Several measures of accuracy were used to evaluate the ANFIS model, to determine the model's capacity to predict the overall rating.

The model was evaluated by four estimates: Mean absolute percentage error (MAPE); root mean squared error (RMSE); mean absolute error (MAE); and coefficient of determination (R^2). The ANFIS model, combined with a subtractive, was used to extract knowledge from user ratings and preferences on item features. This was done by incorporating the element of training into the existing neuro-fuzzy system. The advantage of this method is its flexibility and extendibility, which can be developed for any number of dimensions and criteria of the data set.

From the experiments, it was observed that the proposed method using HOSVD and ANFIS achieved better recommendation accuracy in relation to algorithms in previous work and methods that used SVD and HOSVD exclusively. Experimental results on the movie data set demonstrated the capability of ANFIS modelling using MFs and fuzzy rules without intervention by human experts in multi-criteria collaborative filtering. The experiments confirmed that the hybrid of HOSVD technique and ANFIS, combined with subtractive clustering, significantly improved the predictive accuracy and recommendation quality of multi-criteria collaborative filtering evaluated by the standard accuracy measure.

2.10.4 Accuracy improvement for multi-criteria recommender systems

Jannach, Karakaya and Gedikli (2012) proposed new methods to leverage information derived from multi-dimensional ratings, to improve the predictive accuracy of multi-criteria recommender systems. Support vector regression was used to determine the relative importance of individual criteria ratings, and a suggestion was made to combine user-based and item-based regression models in a weighted approach. Different feature selection strategies were also explored, aside from the automatic adjustment and optimization of the combination weights to further improve the quality of recommendations. Several methods were employed on two data sets (hotel rating and Yahoo! Movies), using different algorithms and quality metrics. The algorithms compared in this paper are SlopeOne, Funk-SVD, MC-Similarity, LS-regress, SV-regress and weighted SVM. Root mean squared error (RMSE), precision and recall were used as evaluation metrics and it was noticed that predictive accuracy increased when data quality increased.

The result achieved by the authors confirmed that the value of detailed customer-provided ratings information can increase recommendation quality, and this quality can be further improved when user-specific and item-specific dependencies in the data are taken into account. The concept of feature selection was applied to multi-criteria recommender systems, and different strategies on how to focus on the most relevant rating dimensions were analysed. The experiments showed that using domain-independent metrics based, for instance, on chi-square statistics and comparably simple procedures can improve prediction accuracy.

2.10.5 Evaluation of recommender systems: A multi-criteria decision-making approach

Sohrabi, Toloo, Moeini and Nalchigar (2015) suggested using a multi-criteria decision-making (MCDM) approach to better model the complexity of evaluating recommender systems. A data envelopment analysis (DEA) approach, a sub-category of MCDM, was proposed to solve the problem of evaluating and selecting recommender systems.

The main idea of DEA is to assess the efficiency of a decision-making unit (DMU) based on its performance of generating output in means of input consumption. The authors showed how different DEA models could help organisational decision-makers evaluate a set of recommender systems and find the best system among given alternatives. DEA models were applied in the evaluation and selection of collaborative filtering recommendation methods in the presence of multiple evaluation metrics, and the data used were obtained from Fouss and Saerens (2008). They calculated four performance criteria (accuracy, robustness, computing time and novelty) for six recommender system algorithms. The first three algorithms were classical memory-based scoring algorithms - basic, binary coefficient (Bin) and cosine coefficient (Cos) – while the last three were kernels on a graph: regularized Lapacian kernel (K_{RL}), commute time kernel (K_{CT}) and Markov diffusion kernel (K_{MD}). Novelty recall, and computing time metrics were calculated on the real MovieLens data set, while robustness of the algorithm was measured on artificially generated data sets. The Charnes, Cooper and Rhode (CCR) and Banker, Chames, Cooper (BCC) models were the two DEA models DEA-Solver applied as input-oriented versions to the data, to differentiate between efficient versus inefficient recommender systems. Thereafter the models proposed by (Toloo, 2012) were applied to further analyse the efficient recommender systems and identify the most efficient one.

The authors concluded that the performance of the six different algorithms, which were compared based on the four criteria (accuracy, computing time, robustness and novelty), identified the commute time kernel (K_{CT}, a kernel-based algorithm), as the most efficient recommender system among the given six alternatives.

2.10.6 Using genetic algorithm for measuring similarity values between users in collaborative filtering recommender systems

Alhijawi and Kilani (2016) proposed a new genetic algorithm-based recommender system, SimGen, which computes similarity values between users without using any of the well-known similarity metric calculation algorithms such as Pearson correlation and vector cosine-based similarity.

SimGen generates an initial random similarity value between every two users and then runs the genetic algorithm (GA). It uses training data to compute the fitness of every generation and testing data to test the resulting similarity, to compute errors in prediction. For the genetic algorithm operator (selection, crossover and mutation), roulette-wheel selection, uniform crossover and single-point mutation techniques were used, and the role of the fitness function was to measure the optimality of the individual (similarity array). Two different data sets were used in the experiment by the authors: Movielens and synthetic data (random biased data to test the recommendation engine).

Traditional evaluation metrics such as mean absolute error (MAE), precision and recall were used to evaluate SimGen. The results of SimGen were compared with those obtained from other approaches using traditional metric on collaborative filtering recommender systems and showed that SimGen was one-and-a-half times faster than a cosine metric, Pearson correlation and other previous work on genetic-based algorithms. It was proved that SimGen achieved high performance in accuracy and quality, compared to current state-of-the-art approaches such as a genetic-based algorithm recommender system, cosine metric and Pearson correlation, in both the synthetic and Movielens data. The authors focused on a single-criterion recommender system that cannot be totally accurate. Our proposed approach used the same techniques; in other words, the use of a genetic algorithm, but on a multi-criteria recommender system, to obtain more accurate and precise predictions.

2.10.7 Improving collaborative filtering recommender system results and performance, using genetic algorithm

Bobadilla, Ortega, Hernando and Alcala (2011) presented a metric to measure similarity between users, which is applicable in collaborative filtering carried out in recommender systems. The proposed metric was formulated via a simple linear combination of values and weights. Values were calculated for each pair of users, between which the similarity was obtained, and weight was calculated just once, making use of a prior stage in which a genetic algorithm extracted weighting from the recommender system which depended on the specific nature of data from each recommender system.

In the genetic algorithm, individuals of the populations were represented in binary forms as strings of 0s and 1s. The initial population was chosen based on the criterion of using a number of individuals in the population which was double the number of bits used to represent each individual. The fitness function was the mean absolute error (MAE) of the recommender system for a particular similarity function, sim_w. For the genetic operators such as selection, crossover and mutation, the chosen method was the fitness proportional selection (the selection probability of an individual depends on its fitness level). One-point crossover and single-point mutation techniques were used, and the stopping criterion was when an individual in the population had a fitness value lower than a constant γ. Three different data sets were used in the experiments – those of Movielens, FilmAffinity and Netflix.

Traditional quality measures (MAE, coverage, precision and recall) were used to evaluate the performance of the proposed algorithm. The results obtained from the proposed algorithms were then compared with those obtained using traditional metrics on a collaborative filtering recommender system (Pearson correlation, cosine and mean squared differences). The result, i.e., similarity functions, obtained by the authors provided significant improvements in prediction quality, recommendation quality and performance, and quicker results than those provided by traditional metrics. The main advantage of this proposed approach is that it can be applied to all collaborative filtering-based recommender systems. These authors applied their approach to a single-criteria recommender system. Ratings from this approach tend not to be accurate, so our proposed approach extends it by focusing on a multi-criteria system to achieve optimum prediction accuracy.

2.10.8 Our solution

A multi-criteria recommender system is still in its infancy, and although several researches have been undertaken to obtain more precise and accurate predictions, further improvements are required to achieve better recommendation accuracy in more complex applications. Our proposed technique attempts to solve this.

Our approach, at its core, tries to understand users' interest efficiently and to recommend to them accurately. The study focuses on the use of an aggregation function where learning will be done using an adaptive genetic algorithm. The rationale behind an adaptive genetic algorithm is that it provides powerful, ideal search techniques to find an optimized solution for a matching item to be recommended to users, out of a large population of variables.

CHAPTER THREE: RESEARCH METHODOLOGY

3.1 Introduction

This chapter presents the outline of the problem we intend to solve and our approach to tackling it. The tools, packages and data set used will be discussed, and we will also introduce analysis of the aggregation function-based technique and adaptive genetic algorithm. This chapter, finally, describes the design of the aggregation function-based adaptive genetic algorithm.

3.2 Multi-criteria recommender system

For a multi-criteria recommender system using the traditional two entities, *users* and *items*, we assume that there is set of all *users* of a system and a set of all possible *items* that can be recommended to the *users*. The utility function R ($u.\ i$) formulation that measures the suitability of recommending an item $i \in Item$ to a user $u \in User$ to estimate the overall ratings R_0 is defined as:

$$\mathrm{R\ (u, i): User \times Item} \qquad \mathrm{R}_0 \qquad (1)$$

The multi-criteria rating recommendation problem extends the traditional single-rating recommendation problem. In measuring the utility function of a user for a given item, an overall rating, $\mathrm{R}_{0,}$ and the user's ratings, R_1, …, R_k for each individual criterion c (c = 1, …, k), are used. The utility function R for the multi-criteria recommendation problem is:

$$R\!: User \times Item \qquad R_0 \times R_1 \times \cdots \times R_k \qquad (2)$$

The multi-criteria recommender system can also employ any of the traditional recommendation techniques, predominantly the content-based filtering, collaborative filtering and hybrid filtering techniques. Our innovative approach focuses on collaborative filtering to capture and model user preferences in a comprehensive manner. Figure 3.1 is a process map of a multi-criteria recommender system.

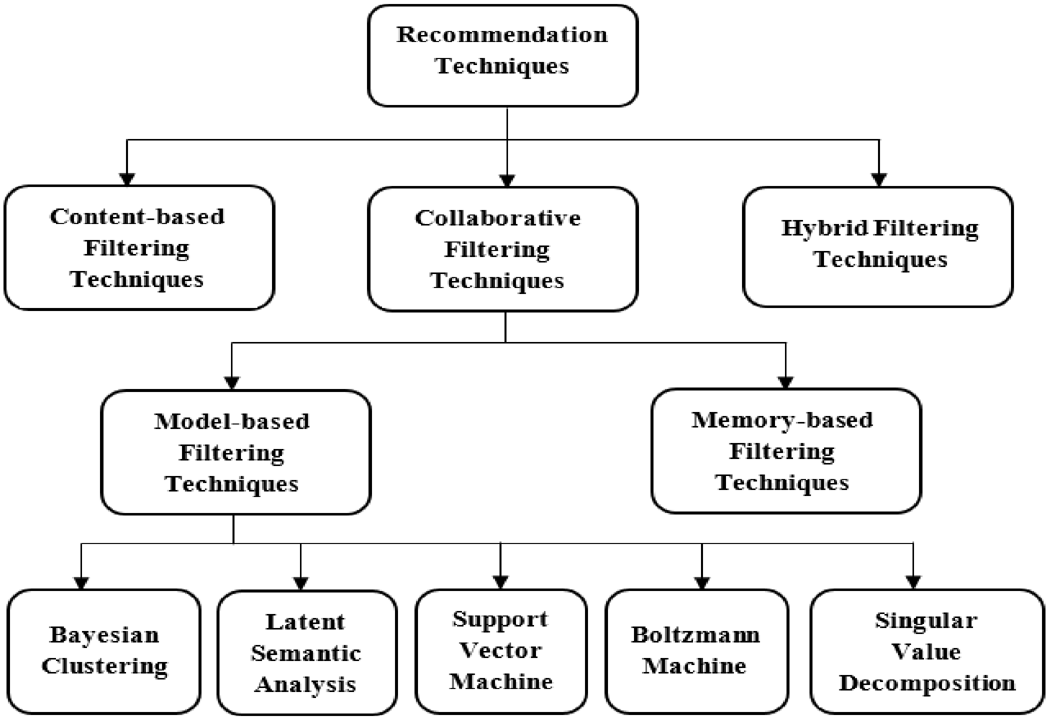

Recommendation Techniques
Content-based Filtering Techniques
Collaborative Filtering Techniques
Hybrid Filtering Techniques
Model-based Filtering Techniques
Memory-based Filtering Techniques
Bayesian Clustering
Latent Semantic Analysis
Support Vector Machine
Boltzmann Machine
Singular Value Decomposition

Figure 3.1: Multi-criteria recommender system

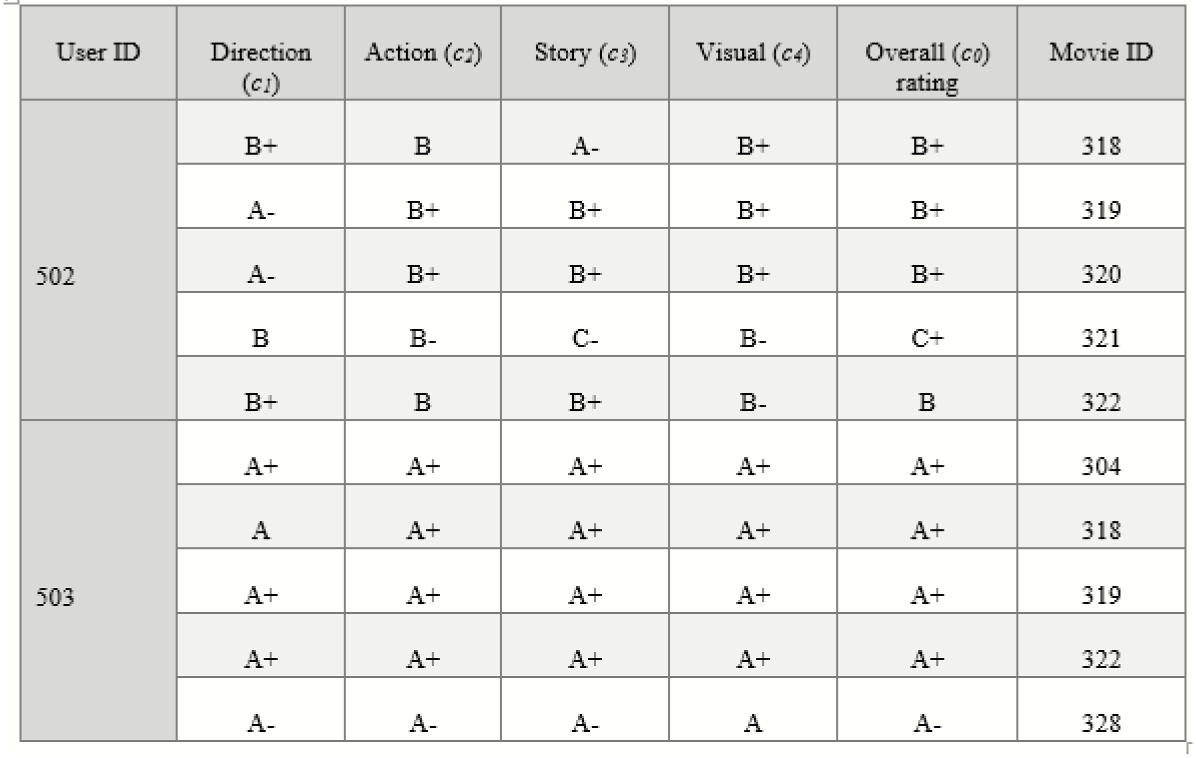
3.3 Data set description

The data set required for this research was retrieved from Yahoo! Movies. The Yahoo! Movies multi-criteria data set contains comprehensive information about user ratings based on four different criteria, i.e. direction (c_1), action (c_2), story (c_3) and visual (c_4). All the ratings were measured in a 13-fold qualitative scale starting with *F*, representing the worst preference, to *A+*, denoting the best preference.

For easy processing purposes, the rating scale (i.e. letters) was transformed into a numerical rating where 1 corresponded to *F*, the worst preference, and 13 corresponded to *A+*, the best.

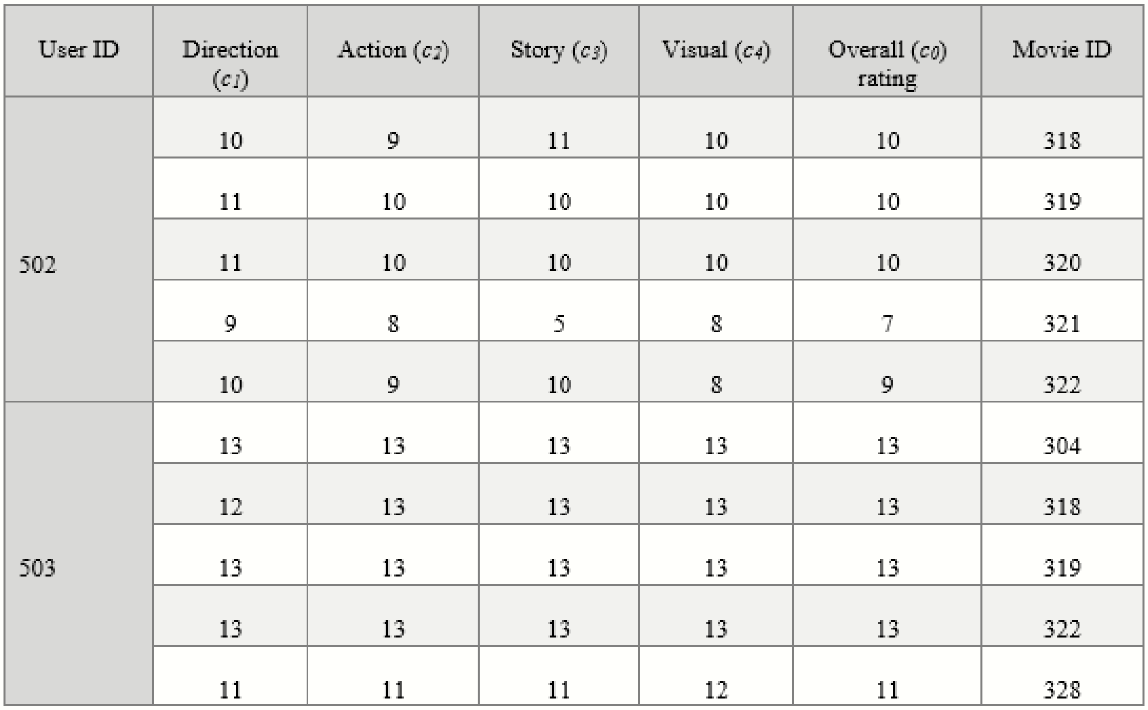
In addition to the individual criteria ratings, there is an overall (c_0) rating that measures the global preferences for each of the movies. Table 3.1 shows typical raw data extracted directly from the Yahoo! Movies website. Table 3.2 shows the same data transformed to their equivalent in numbers.

Table 3.1: Sample of multi-criteria raw data input matrix before conversion



User ID	Direction (c_1)	Action (c_2)	Story (c_3)	Visual (c_4)	Overall (c_0) rating	Movie ID
502	B+	B	A-	B+	B+	318
	A-	B+	B+	B+	B+	319
	A-	B+	B+	B+	B+	320
	B	B-	C-	B-	C+	321
	B+	B	B+	B-	B	322
503	A+	A+	A+	A+	A+	304
	A	A+	A+	A+	A+	318
	A+	A+	A+	A+	A+	319
	A+	A+	A+	A+	A+	322
	A-	A-	A-	A	A-	328

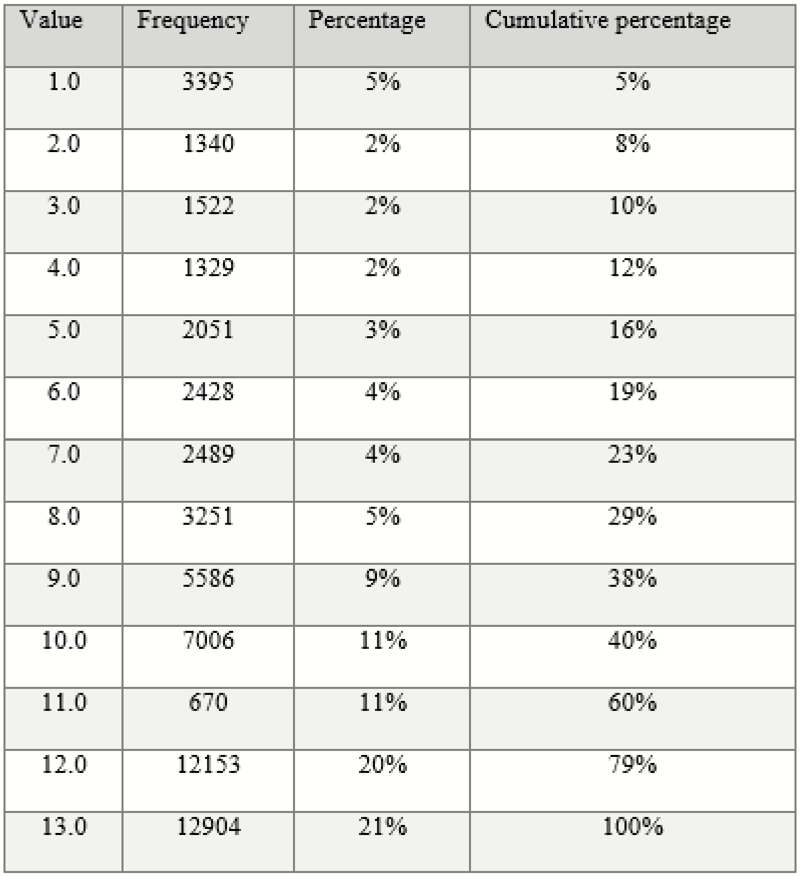
Table 3.2: Sample of multi-criteria raw data input matrix after conversion



User ID	Direction (c_1)	Action (c_2)	Story (c_3)	Visual (c_4)	Overall (c_0) rating	Movie ID
502	10	9	11	10	10	318
	11	10	10	10	10	319
	11	10	10	10	10	320
	9	8	5	8	7	321
	10	9	10	8	9	322
503	13	13	13	13	13	304
	12	13	13	13	13	318
	13	13	13	13	13	319
	13	13	13	13	13	322
	11	11	11	12	11	328

Data cleaning was done on the data set to detect and remove inconsistencies, i.e. to remove any case of missing ratings for at least one or more of the four criteria and the overall ratings, and to improve the quality of the data set. To ensure an adequate set of evaluated items for each user, subsequent filters were applied to remove users with less than five rated movies. To this end, the training set each time consisted of at least five movies and the resulting experimental data set included 6 078 different users and 976 movies, which gives a total of 62 156 ratings with a sparsity of 0.0105. The global average, median and standard deviation for the direction, action, story, visual and overall rating in four decimal points are 9.5221, 11.0000, and 3.5232 respectively. Table 3.3 shows the rating matrix of the multi-criteria recommender system data set.

Table 3.3: Rating matrix of the data set



Value	Frequency	Percentage	Cumulative percentage
1.0	3395	5%	5%
2.0	1340	2%	8%
3.0	1522	2%	10%
4.0	1329	2%	12%
5.0	2051	3%	16%
6.0	2428	4%	19%
7.0	2489	4%	23%
8.0	3251	5%	29%
9.0	5586	9%	38%
10.0	7006	11%	40%
11.0	670	11%	60%
12.0	12153	20%	79%
13.0	12904	21%	100%

3.4 Choice of programming language

A multi-criteria recommender system based on adaptive genetic algorithm was implemented using Java language. Java is an object-oriented programming language developed by Sun Microsystems. It was designed to be small, simple and portable across platforms and operating systems. Java will be used in our research for transforming and extracting information from the Yahoo! Movies data set. Java as a language has significant advantages (it is platform-independent, simple, robust, secure, portable and dynamic) over other programming languages, which makes it suitable for just about any programming task.

3.5 Proposed system

The proposed system creates a recommendation system that effectively makes users decision-distinct, for example a movie recommender system where users provide the recommender system with a multi-criteria rating (between 1 and 13) for each movie they have seen. Suppose a model-based collaborative filtering approach is used for the rating prediction. If a user, *Jamal*, wants to rate a movie, *Finding Dory,* on some specific dimensions such as story, visual effects, direction and acting etc, then he may give his rating on each of the dimensions based on priority that can be in decreasing order, i.e. 12 for story, 10 for visuals, 9 for direction and 7 for acting. The overall rating of an item is not just another rating independent of others but serves some "aggregation" function of the multi-criteria rating of the item. This is shown in equation (3).

$$R_0 = f(R_1, \dots, R_k) \qquad (3)$$

If the story's criteria rating are set to be the "highest priority" by the movie recommendation application, i.e., the movies with high story ratings are well liked overall by some users regardless of other criteria ratings. Since *Jamal's* rating for the story criteria is high, the overall rating of the movie, *Finding Dory*, must also be predicted high in order to be accurate. In a case where a movie is to be recommended to *Jamal*, it may be difficult to accurately recommend a movie because there can be conflicting and competing ratings.

Our proposed approach for a rating estimation was portioned into three steps. The first step was to decompose the *k*-dimensional multi-criteria rating space into a k single-rating recommendation problem, and any single criterion recommendation technique (collaborative filtering, content-based, hybrid, etc.) can be used to estimate ratings for each individual criterion. Secondly, we used machine learning techniques to estimate the aggregation function based on the known ratings. Thirdly, the estimated multi-criteria ratings in step 1 and the function estimated in step 2 were used directly to calculate the predicted overall rating.

3.5.1 Predicting *N* multi-criteria ratings

The k-dimensional multi-criteria ratings were decomposed into single ratings, converting the problem into a k single-criteria rating recommendation problem, as mentioned earlier. In other words: The multi-criteria rating recommendation problem, $R\!: User \times Item \longrightarrow R_0 \times R_1 \times \cdots \times R_k$ is converted to a k single-rating recommendation problem: $R\!: User \times Item \longrightarrow R_c$ (c = 1, …, k). Collaborative filtering, the most popularly used filtering technique, was applied to predict these unknown individual criteria ratings. Asymmetric singular-value decomposition was used to predict unknown individual criteria.

3.5.2 Asymmetric singular-value decomposition (ASVD)

ASVD is an extension of the traditional singular-value decomposition (SVD) discussed earlier in this project. The SVD model associates each item i with an item-feature vector and each user u with a user-feature vector.

The prediction of an item to a user is given in equation (4):

$$R'_{ui} = b_u + b_i + \mu + L_i{}^T M_u \qquad (4)$$

ASVD, on the other hand, is associated with three vectors ($c_i, d_i, L_i \in \mathbb{R}^f$) and the users are represented through the items that they prefer. Thus, the user feature M_u used in SVD is replaced by the sum $|R(u)|^{-0.5}\sum_{j\in R(u)}(R_{uj} - b_{uj})c_j + |N(u)|^{-0.5}\sum_{j\in N(u)} d_j$ suggested by Bokde et al. (2014). Thus, we have our prediction of an item to a user as given below:

$$R'_{ui} = b_u + b_i + \mu + L_i{}^T(|R(u)|^{-0.5}\sum_{j\in R(u)}\left(R_{uj} - b_{uj}\right)c_j + |N(u)|^{-0.5}\sum_{j\in N(u)} d_j) \qquad (5)$$

We identified the values of the parameters involved by minimizing the regularized squared error function associated with (5) as shown below:

$$\min_{b^*,L^*,{c_i}^*,{d_i}^*} \sum_{(u,i)\in K} (R_{ui} - \mu - b_u - b_i - {L_i}^T (|R(u)|^{-0.5} \sum_{j\in R(u)} \left(R_{uj} - b_{uj}\right) c_j + |N(u)|^{-0.5} \sum_{j\in N(u)} d_j))^2 + \lambda (b_u^2 + b_i^2 + \| L_i \|^2 + \sum_{j\in R(u)} \| c_i \|^2 + \sum_{j\in R(u)} \| d_i \|^2) \qquad (6)$$

Where R_{ui} is the rating given by user u to item i, μ is the global average rating, $b_u$and b_i indicate the observed deviation of user u and item i respectively, $R(u)$ is the set of all items rated by the user u which are known ratings, $N(u)$ is the set of all items rated by the user u which are either known or unknown ratings, i.e., which user u provided an implicit preference, and j denotes an item.

Stochastic gradient descent was employed for minimizing the regularized squared error. The parameters γ and λ are assigned small positive real numbers such as 0.004 and 0.02 respectively.

The next step after obtaining individual predicted criteria ratings was to aggregate these individual ratings to get an overall rating, which is very important because of users' different priority choices on diverse criteria.

3.5.3 Learning the function

This step is targeted at estimating the relationship f between the overall rating and the underlying multi-criteria ratings of items, such that $R_0 = f(R_1, \dots, R_k)$.

The optimum weight for an individual criterion, for each user over various criteria, was obtained through genetic algorithm. Genetic algorithm is an efficient and random search algorithm based on natural evaluation and genetic inheritance, which combines exploitation of past results with the exploration of new areas of the search space. It provides a global optimum result with an inherent implicit parallelism that adaptively adjusts the search direction. A genetic algorithm was employed in our approach, mainly to yield the best fitted chromosomes that can represent the weights provided by the user to each dimension. Every user gives a different priority to each feature, which can be referred to as feature weight. The adaptive genetic algorithm in our approach adapts the feature weights to capture the user priorities for different features.

The feature weights of user u_a is represented as a set of weights, weight (u_a) = [w_i] (where i = 1, …, z, and z is the number of features). A feature is ignored for further calculation when the weight of that feature is zero, thus enabling feature selection to be adaptive to each user's preferences. The genotype is a string of double-valued vectors, i.e. the chromosome representation is done using double-valued vectors. The proposed adaptive genetic algorithm (adaptive GA) parameters are discussed as follows:

***Initial population*:** Genetic algorithm searches many points in the search space. The search space used in this study consists of valid feature weights. The algorithm allows the experiment to define the size of the population, so the initial population is generated randomly with a population of 40.

***Fitness function*:** This is the most important aspect of a genetic algorithm and measures the level of success of each chromosome after the initialization of the population. Calculating the fitness function is a challenging task and not a trivial one for this application. In our proposed approach, fitness function is calculated using linear regression. For an active user, each chromosome c is assigned a fitness function and is tested. The fitness function measures the prediction accuracy of items using the randomly generated weights [w_i] (where i = 1, …, z, and z is the number of features which is four (4)) as defined in the current chromosome.

Our proposed fitness function is formulated as follow:

$$fitness\ (c) = \ 1 - \left(\frac{\sum_{j=1}^{N_a} \left| R_{0j} - \left(\frac{w_1 R_{j1} + \cdots + w_z R_{jz}}{(w_1 + \cdots + w_z)} \right) \right|}{N_a} \right) \qquad (7)$$

Where N_a is the total number of times each chromosome c is tested, R_j is the actual ratings, and R_{0j} is the overall ratings of user u_a to item i_j. Our aim is to obtain the most optimum weights to use for predicting the overall unknown ratings.

***Reproduction*:** The reproduction process is basically based on the values of each individual's fitness function. The idea is that any individual with a higher fitness value will have a higher chance of surviving to the next generation. Fitness function of every individual in a population is calculated and the ones with the highest fitness value are selected. The best individuals are kept for the next generation, which will in the course of the genetic process modify the population. It is a competition between the parents and children.

***Selection*:** Using the roulette wheel selection method, parents are selected according to their fitness. Better chromosomes have more chances of being selected. In this method, each individual is assigned a slice of the circular roulette wheel, and the size of the slice is proportional to the individual fitness of the chromosomes. The wheel is then spun and any individual who owns the selection on which the wheel lands each time is chosen.

***Crossover*:** The traditional single-point crossover was used. A crossover point was randomly selected along the length of the mated strings and then the bits next to the crossover point were exchanged. The crossover probability was based on the fitness function, because it is an adaptive genetic algorithm.

Mutation: A uniform mutation is adopted in order to introduce diversity. This mutation operator replaces the value of the chosen gene with a uniform random value selected between the specified upper and lower bounds for the gene. Since it is an adaptive genetic algorithm, the mutation probability was generated based on the fitness function, so it is not fixed.

Termination condition: The termination condition was based on a defined number of generations (100) and also when there is convergence of the best-fit chromosomes.

Figures 3.2 and 3.3 that follow are flow charts of our adaptive genetic algorithm and the adaptive genetic algorithm respectively.

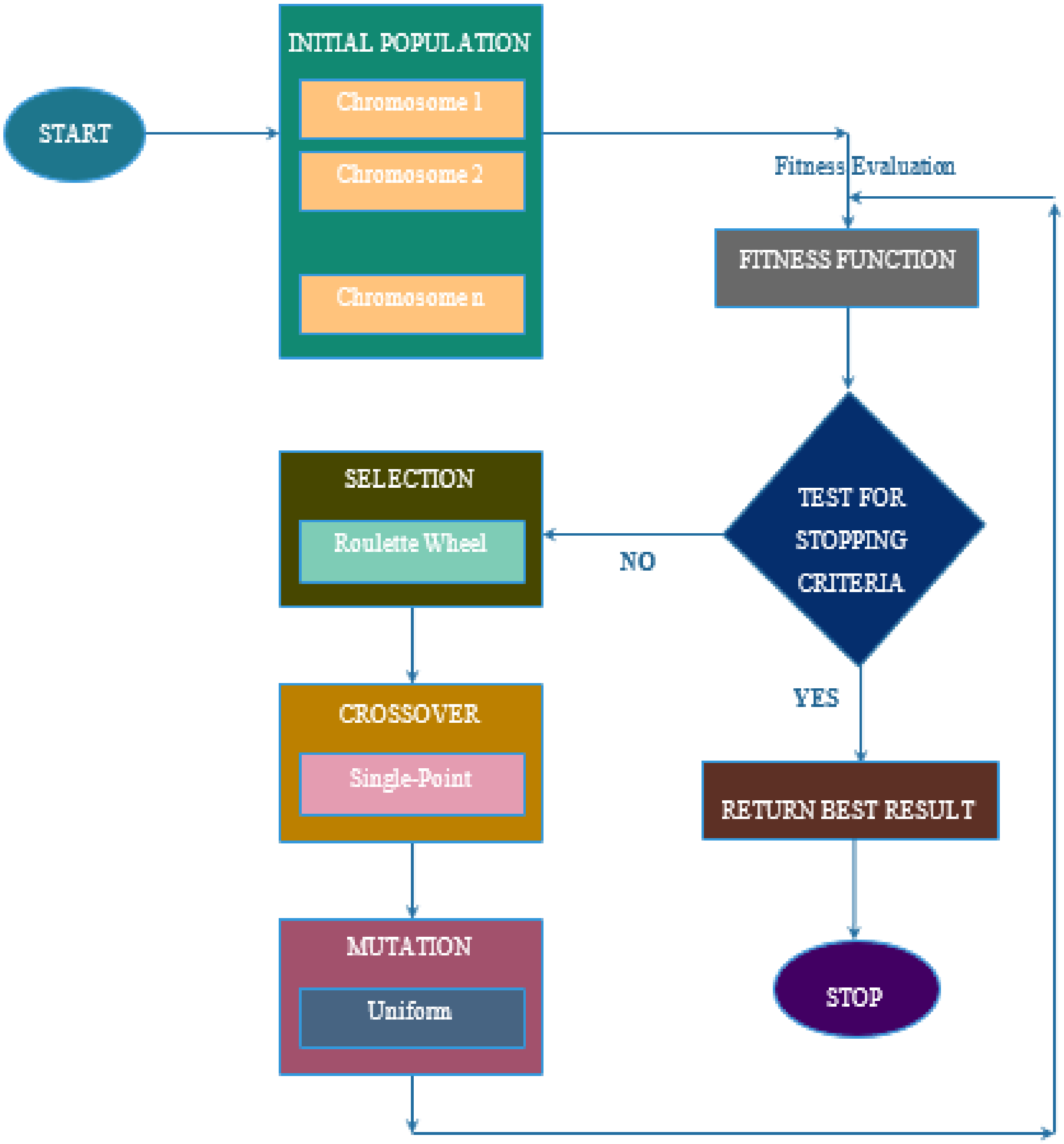


Figure 3.2: Process diagram of an Adaptive Genetic Algorithm

```
Algorithm 1: Adaptive Genetic Algorithm

Adaptive Genetic Algorithm ()
{
        Randomly create an initial population from our search space
        Calculate the fitness value of each individual in the population
        While not termination criteria do
        {
                Select the best-fit individuals from the population for reproduction
                Perform crossover to breed new individuals
                Ascertain if the new individuals are feasible and unique
                Perform mutation to generate new individuals
                Ascertain if the new individuals are feasible and unique
                Evaluate the individual fitness of new individuals
                Keep the best-fit individuals in the population and eliminate the worst ones.
        }
}
```

Figure 3.3: Adaptive Genetic Algorithm

3.5.4 Predicting the overall rating

The unknown overall rating R'_0 in the testing data is finally computed using the weighted sum of individual predicted ratings generated.

$$R'_0 = \frac{\sum_{i=1}^{z} w'_i \times R'_{ji}}{\sum_{i=1}^{z} w'_1} \qquad (8)$$

R'_j is the predicted rating of user u_a to item i_j in the multi-criteria rating. Expanding (8) will give the formula below:

$$R'_0 = \frac{w'_1 R'_{j1} + \cdots + w'_z R'_{jz}}{(w'_1 + \cdots + w'_z)} \qquad (9)$$

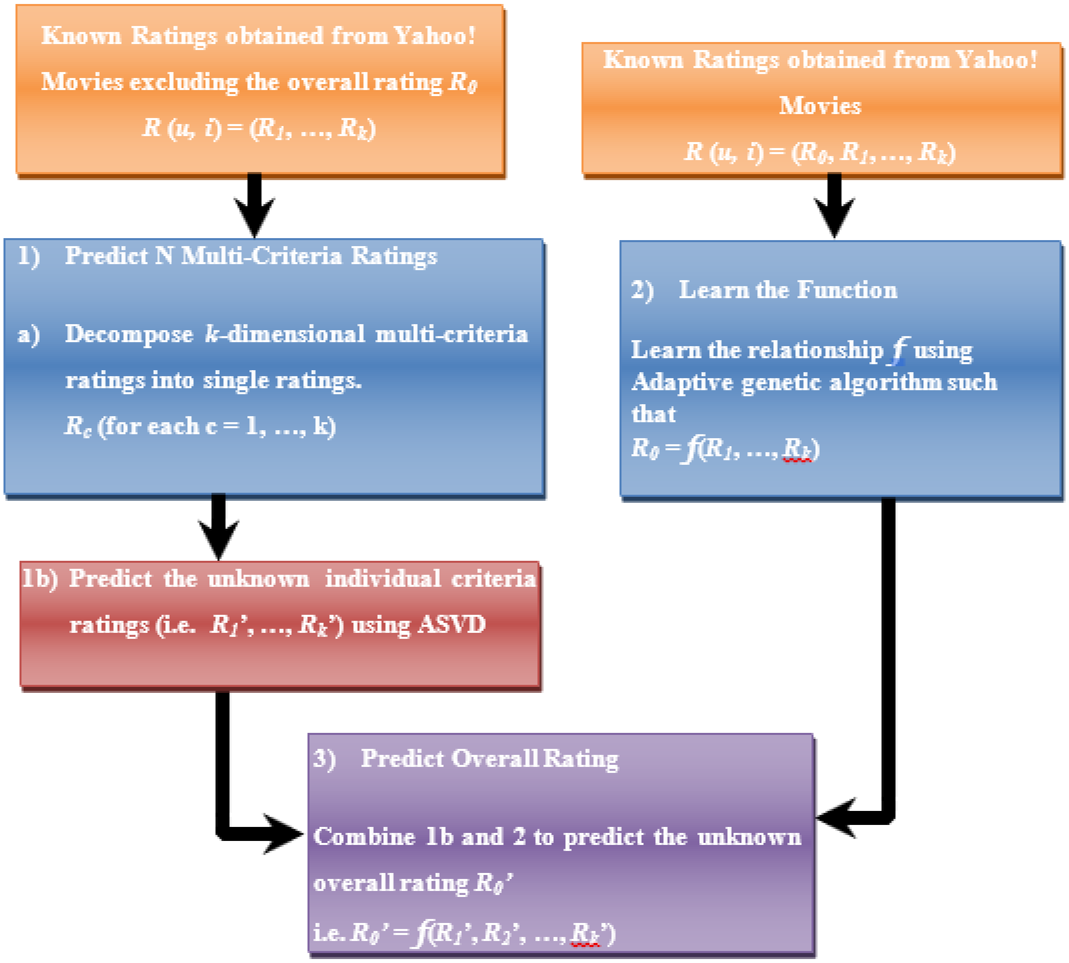

Known Ratings obtained from Yahoo! Movies excluding the overall rating R_0
$R(u, i) = (R_1, \ldots, R_k)$
Known Ratings obtained from Yahoo! Movies
$R(u, i) = (R_0, R_1, \ldots, R_k)$
1) Predict N Multi-Criteria Ratings
a) Decompose k-dimensional multi-criteria ratings into single ratings.
R_c (for each c = 1, …, k)
2) Learn the Function
Learn the relationship f using Adaptive genetic algorithm such that
$R_0 = f(R_1, \ldots, R_k)$
1b) Predict the unknown individual criteria ratings (i.e. $R_1', \ldots, R_k'$) using ASVD
3) Predict Overall Rating
Combine 1b and 2 to predict the unknown overall rating R_0'
i.e. $R_0' = f(R_1', R_2', \ldots, R_k')$

Figure 3.4: Framework of the proposed Multi-criteria Recommender system using Adaptive Genetic Algorithm

```
Algorithm 2: Pseudo-code of the proposed approach

Input: Multi-criteria dataset
Output: Accurate Items prediction to active users
Begin algorithm
    Phase 1: Multi-criteria dataset pre-processing
        Step 1: Dataset pre-processing to detect and remove inconsistencies
        Step 2: Subsequently filter to remove users with less than five rated movies
                    $R\ (user, item) = (R_0, R_1, \dots, R_k)$
    Phase 2: Aggregation function approach
        Step 1: Predict N Multi-criteria ratings
            Process 1: Decompose the k-dimensional multi-criteria ratings into single-criterion ratings.
                    $R: User \times Item \longrightarrow R_0 \times R_1 \times \cdots \times R_k$
                    $R: User \times Item \longrightarrow R_c\ (c = 1, \dots, k)$
            Process 2: Use collaborative filtering technique to predict the individual unknown ratings
                    $R_c'$
        Step 2: Learn the function
            Process 1: Use Adaptive genetic algorithm to learn the appropriate relationship
            Process 2: Estimate relationship $f$ such that $R_0 = f(R_1, R_2, \dots, R_k)$
        Step 3: Predict Overall ratings
            Process 1: Predict the overall rating using the predicted criterion rating $R_c'$ and the
                    estimated appropriate relationship $f$.
            Process 2: Compute overall rating $R_0' = f(R_1', R_2, \dots, R_k')$
    Phase 3: Recommendation
        Step 1: Generate the List of the Top-N recommendations
        Step 2: Select the first N-items as the Top-N recommended set to the active user
End algorithm
```

Figure 3.5: Algorithm of the proposed approach

The goal of the multi-criteria recommendation system is to produce a recommendation list for an active user. The recommendation system predicts the unknown overall ratings using the feature weight function estimate and the multi-criteria value estimated.

Finally, all the unrated items are sorted in non-increasing order with respect to the overall ratings, which is the list of the Top-N recommendations and the first N-items are selected as the Top-N recommended set.

CHAPTER FOUR: IMPLEMENTATION

4.1 Introduction

This chapter presents the detailed implementation of the proposed system using the adaptive genetic algorithm. In order to prove our methodology works, we implemented it in a feasible manner, ran, generated results and compared them with traditional collaborative filtering. In this chapter, methods that are well accepted were used to evaluate the effectiveness of the algorithms and the result was discussed.

4.2 Performance evaluation

The performance evaluation of a machine learning system is an important task for result validation and comparing results with other similar solutions. Many performance metrics exist, but it is necessary to select the most appropriate measure for your problem. Thus, to test the performance of our multi-criteria recommender based adaptive genetic algorithm, four main evaluation metrics methods were used to measure the prediction accuracy as follows:

***Mean Absolute Error (MAE)*:** This measure how close our predicted ratings are to the actual outcome. The MAE of the system was calculated using equation (1).

$$MAE = \sum_{j=1}^{N_t} \left| \frac{(R\prime_0 - R_{0(Overall)})}{N_t} \right| \quad (1)$$

Where N_t is the number of test sets, R'_0 is the predicted ratings generated from the test sets, and $R_{o(Overall)}$ is the actual rating from the test sets. The test set consists of different users' ratings to different movies.

Root Mean Square Error (RMSE)**:** RMSE was also used to measure the rating prediction accuracy, where N_t is the number of test sets, R'_0 is the predicted ratings generated from the test sets, $R_{o(Overall)}$ is the actual ratings from the test sets as shown below in equation (2) below.

$$RMSE = \sqrt{\sum_{j=1}^{N_t} \frac{\left(R\prime_0 - R_{o(Overall)}\right)^2}{|N_t|}} \qquad (2)$$

Area under the Curve (AUC)**:** the AUC of a receiver operating characteristics (ROC) curve is used for measuring our ranking accuracy and it is useful for comparing algorithms independently of application. For each user *u*, finding the AUC of the sensitivity rate against specificity measures how accurate the algorithms separate predictions into relevant and irrelevant items. The AUC of the system was calculated using equation (3).

$$AUC_u = \frac{1}{N}\left[\left(\sum_{i=1}^{tp_u} rank_{ui}\right) + \left(\frac{tp_u+1}{2}\right)\right] \qquad (3)$$

Where *N* is the length of the recommended list, $rank_{ui}$ is the position of the *kth* relevant item among the list of *N* recommended items, and tp_u is the number of true positive of each user *u*.

Mean Average Precision (MAP)**:** MAP is used to measure classification accuracy and it computes the Average Precision (AP) across several different levels of recall. The result of MAP should be close to 1 for a good algorithm and 0.5 for a bad algorithm. MAP of the system was calculated using the equation (5).

$$AP = \frac{\sum_{i=1}^{N}(precision(i) \times recall(i))}{number\ of\ relevant\ items} \qquad (4)$$

$$MAP = \frac{\sum_{u=1}^{M} AP_u}{M} \qquad (5)$$

Where *M* is the total number of relevant items among the list of Top-*N* recommended items.

Mean Reciprocal Rank (MRR)**:** MRR is used to measure the ranking accuracy of the system by finding a single resource for a given query. MRR is the mean of the multiplicative inverse of the rank of the first correct item, i.e., it focuses on the single highest-ranked relevant item. MRR of the system was calculated using equation (6).

$$MRR = \frac{1}{N}\left(\sum_{i=1}^{N} \frac{1}{rank_i}\right) \qquad (6)$$

Where N is the total number of relevant items among the list of Top-N recommended items, $rank_i$ is the rank position of the first relevant item for the *i-th* query.

Fraction of Concordant Pairs (FCP): FCP is used to measure the ranking accuracy of the system. FCP goes through each pair of items and their ranks and check if they are ranked correctly against each other, i.e., it checks the fraction of all pairs that it puts in the correct order. FCP of the system was calculated using equation (7).

$$FCP = \frac{a_c}{a_c - a_d} \qquad (7)$$

Where a_c is the number of concordant pairs defined as $a_c = \sum_{u \in U} |(i.j)| \{R'_{ui} > R'_{uj} \Rightarrow R_{ui} > R_{uj}\}$, and a_d is the corresponding number of discordant pairs calculated as $a_d = \sum_{u \in U} |(i.j)| \{R'_{ui} > R'_{uj} \Rightarrow R_{ui} \leq R_{uj}\}$. The predicted R'_{ui} and R'_{uj} are concordant pairs for some items *i* and *j* such that when $R'_{ui} > R'_{uj}$, then their corresponding ratings R_{ui} and R_{uj} from the dataset must also satisfy the same condition $R_{ui} > R_{uj}$; or else the items i and j are called discordant pairs.

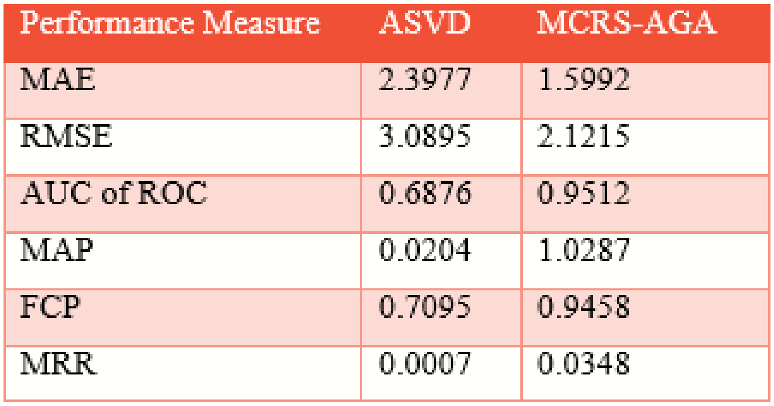
4.3 Result and discussion

Our experiment was carried out using Yahoo! Movies dataset. To ensure the reliability of our proposed approach, we used a standard 10-folds cross-validation technique, where the dataset is randomly divided into 10 disjoint subsets. Nine-tenths of the data set was used for training and the remaining one-tenth for testing the rating prediction. This process was repeated 10 times and evaluation was performed on the predicted ratings.

Furthermore, to demonstrate the effectiveness and feasibility of the proposed approach, the overall rating computed with the traditional collaborative filtering technique (ASVD) was compared with the proposed multi-criteria recommender system based on the adaptive genetic algorithm (MCRS-AGA) using various performance measures. To investigate and analyse the predictive performance of the proposed approach, six (6) different evaluation metrics were used to measure the prediction, ranking and usage accuracy of the system, which includes MAE, RSME, AUC of the ROC, MRR, MAP and FCP.

The generated MAE, RSME, AUC of the ROC, MAP, MRR and FCP from the experiment shown in Table 4.1 proved that the proposed multi-criteria recommender system based on the adaptive genetic algorithm is more accurate than the corresponding traditional single-criteria recommender system. The results confirmed that our proposed approach achieved better predictive performance in terms of a decrease in prediction errors (MAE and RMSE), increase in classification accuracy (MAP), and increase in the ranking accuracy (AUC, MRR and FCP) as compared to the traditional ASVD.

Table 4.1: Performance Evaluation Results



Performance Measure	ASVD	MCRS-AGA
MAE	2.3977	1.5992
RMSE	3.0895	2.1215
AUC of ROC	0.6876	0.9512
MAP	0.0204	1.0287
FCP	0.7095	0.9458
MRR	0.0007	0.0348

Figure 4.1 is a column chart that shows a clear difference between our proposed algorithm (MCRS-AGA) and the traditional ASVD. The MAE and RMSE show a decrease in prediction errors; and AUC of ROC, MAP, FCP, and MRR show an increase in both classification and ranking accuracy.

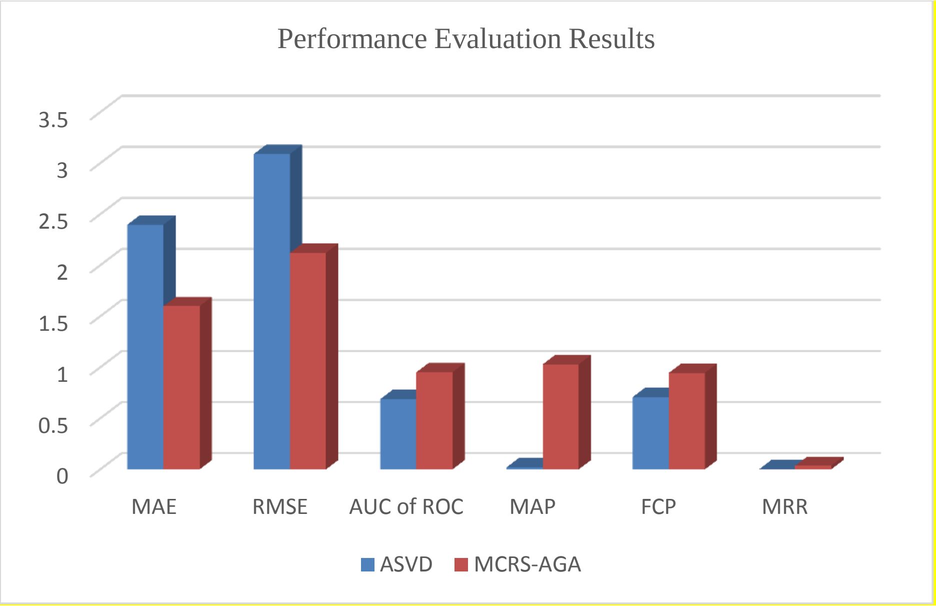
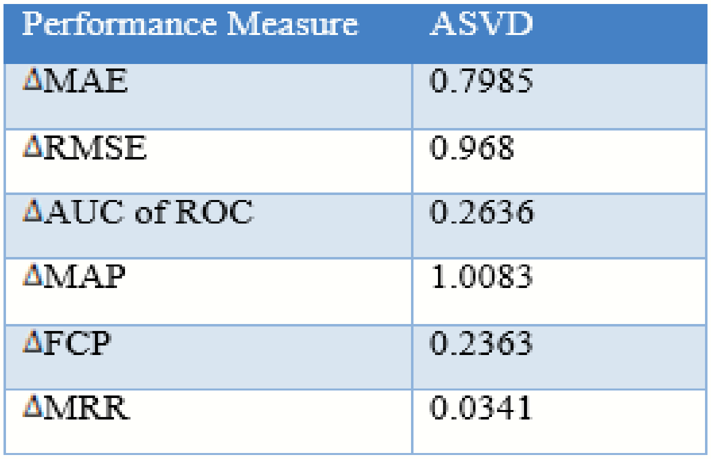


Figure 4.1: Performance evaluation results

The numerical difference in accuracy between our proposed multi-criteria recommender systems based on adaptive genetic algorithm and traditional single-criterion rating (ASVD) is shown in Table 4.2. ΔMAE and ΔRMSE indicates the decrease in prediction error between the MCRS-AGA and ASVD, ΔMAP indicates increase in classification accuracy between the MCRS-AGA and ASVD, and ΔAUCof ROC, ΔFCP and ΔMRR indicates an increase in ranking accuracy between the MCRS-AGA and ASVD.

Table 4.2: Difference in Accuracy between MCRS-AGA and ASVD



Performance Measure	ASVD
ΔMAE	0.7985
ΔRMSE	0.968
ΔAUC of ROC	0.2636
ΔMAP	1.0083
ΔFCP	0.2363
ΔMRR	0.0341

4.4 Conclusion

In conclusion, based on the experiment and results, it is obvious that the multi-criteria recommender system based on an adaptive genetic algorithm provided better predictive performance than the traditional single-criterion rating system (ASVD) in terms of a decrease in prediction errors and an increase in classification and ranking accuracy.

CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Introduction

This chapter summarizes the work conducted within this thesis, and the contributions of this work are highlighted. It also concludes the work done with discussion on future work and provides recommendations.

5.2 Summary and contributions

The most critical problems of a multi-criteria recommender system are prediction accuracy and multi-criteria optimization. In a multi-criteria rating system a user rates an item giving priority to some specific dimensions over others, and this varies from user to user according to each user's personal interest. Thus, it is important to find a suitable relationship between the individual criteria ratings and overall ratings, because each user has different priorities on various dimensions of an item.

This thesis was aimed at determining the impact of an adaptive genetic algorithm in enhancing the prediction accuracy of a multi-criteria recommender system. In our approach we treated the multi-criteria problem as an optimization problem and applied a weighted sum average which was achieved by combining values derived from an adaptive genetic algorithm. To achieve this, a model-based approach focusing on the use of the aggregation function-based technique was used. The aggregation function-based technique consists of mainly three steps after the data acquisition.

The first step was the prediction of N-multi-criteria ratings. The k-dimensional multi-criteria rating was decomposed into individual ratings and a single rating collaborative filtering recommender system approach, i.e. asymmetric singular value decomposition (ASVD) technique was applied to predict individual unknown ratings. The aggregation of individual predicted ratings to achieve an overall rating is one of the most important tasks, because of users' different priority choices on the various dimensions.

The second step was to learn the function. An optimized weighting value was derived for each individual criterion for each user, using an adaptive genetic algorithm. These weights in essence are more like showing the level of priorities each user rated an item on. The use of an adaptive genetic algorithm provided significant performance improvement.

Finally, the prediction of the overall rating was achieved by using the weighted sum of the individual predicted ratings generated from the first step and the second step. Several experiments were carried out on the Yahoo! Movies data set, of which our proposed multi-criteria recommender system based on an adaptive genetic algorithm provided better accuracy. In addition, the investigation from the experimental result showed that the proposed multi-criteria recommender system based on an adaptive genetic algorithm produced the highest prediction accuracy, compared to the traditional collaborative filtering recommendation system.

Our main contributions include a formulated novel algorithm which treated the multi-criteria recommendation problem as an optimization problem and also a developed system proficient enough to recommend the most appropriate item to a user. An adaptive genetic algorithm model was formulated which was used to model multi-criteria recommendation problems. The adaptive genetic algorithm helped in getting the relation between overall rating and individual criterion rating in a multi-criteria data set based on derived weighting values.

5.3 Conclusion

The proposed approach had a number of aims and objectives as discussed in Chapter One of the thesis. Among the objectives set out, and from the analysis of the implementation methodology, we were able to develop a novel adaptive genetic algorithm in which the crossover rate and mutation rate were based on the fitness function, which provided significant performance improvements over standard implementation to generate better potential solutions.

The developed adaptive genetic algorithm was then used to solve a multi-criteria recommendation problem by generating optimum weights whereby a user rated an item based on the priority of item criteria. The weighting was aggregated with the predicted ratings generated using one of the traditional recommendation techniques to achieve the overall ratings. This approach bridged the gap between each individual criterion rating and the overall rating.

The proposed approach achieved the aim of providing predictive performance, including a decrease in prediction errors, an increase in classification and ranking accuracy and obtaining a high correlation between predicted and actual values. Finally, experimental results obtained from comparing the predictive performance of our multi-criteria recommender technique using adaptive genetic algorithm with the existing traditional collaborative filtering approach clearly indicated that our proposed scheme yielded better results and outperformed traditional collaborative filtering recommender systems.

5.4 Recommendation and future work

Recommender systems are now widely being employed in various types of application and domains (such as e-commerce, e-business, social media, entertainment etc.) and there are many possibilities to further continue to improve the algorithm to increase prediction accuracy and efficiency. A potential researcher who might be interested in this topic in the near future should use a different stochastic or evolutionary algorithm to improve the prediction accuracy of the multi-criteria recommender system.

In future we plan to extend the current approach by employing a hybrid of an adaptive genetic algorithm and fuzzy logic techniques on more than one real-world data set to learn appropriate aggregation functions for combining individual criterion ratings.

REFERENCE

Abdullah, S., & Turabieh, H. (2008). Generating university course timetable using Genetic Algorithms and local search. In *Proceedings - 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008* (Vol. 1, pp. 254–260). https://doi.org/10.1109/ICCIT.2008.379

Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, *22*(3), 48–55. https://doi.org/10.1109/MIS.2007.58

Adomavicius, G., & Tuzhilin, a. (2005). Toward the Next Generation of recommender systems: a Survey of the State of the Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99

Adomaviciuszan, G., & Tuzhilin, H. (2014). *INFORMS Tutorials in Operations Research Personalization and recommender systems*. https://doi.org/10.1287/educ.1080.0044

Alhijawi, B. (2016). Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender ... 8VLQJ * HQHWLF $ OJRULWKPV IRU 0HDVXULQJ WKH 6LPLODULW \ 9DOXHV EHWZHHQ 8VHUV LQ & ROODERUDWLYH, (June). https://doi.org/10.1109/ICIS.2016.7550751

Bobadilla, J., Ortega, F., Hernando, A., & Alcal??, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-Based Systems*, *24*(8), 1310–1316. https://doi.org/10.1016/j.knosys.2011.06.005

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012

Bokde, D., Girase, S., & Mukhopadhyay, D. (2014). Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey. *International Journal of Advance Foundation* *and Research in Computer (IJAFRC)*, *1*(6).

Burke, R. (2002). Hybrid recommender systems : Survey and. *User Modeling and UserAdapted Interaction*, *12*(4), 331–370. https://doi.org/10.1023/A:1021240730564]

Candillier, L., Jack, K., Fessant, F., & Meyer, F. (1997). *recommender systems*. *Communications of the ACM* (Vol. 40). https://doi.org/10.1145/245108.245121

Chakraborty, R. (2010). Fundamentals of Genetic Algorithms. *Thesis on GA 42*, (x).

Eiben, A. E., & Smith, J. E. (2007). *Introduction to evolutionary computing*. *With 28 tables*. https://doi.org/10.1007/978-3-662-05094-1

Fouss, F., & Saerens, M. (2008). Evaluating Performance of recommender systems: An Experimental Comparison. *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, *1*, 8–11. https://doi.org/10.1109/WIIAT.2008.252

Goodman, E. D. (2008). Introduction to genetic algorithms. *Proceedings of the 2008 GECCO Conference Companion on Genetic and Evolutionary Computation - GECCO '08*, *2*, 2277. https://doi.org/10.1145/1388969.1389056

Hassan, M., & Hamada, M. (2017). A Neural Networks Approach for Improving the Accuracy of Multi-Criteria recommender systems. *Applied Sciences*, *7*(9), 868. https://doi.org/10.3390/app7090868

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5–53. https://doi.org/10.1145/963770.963772

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261–273.

https://doi.org/10.1016/j.eij.2015.06.005

Jannach, D., Karakaya, Z., & Gedikli, F. (2012). Accuracy Improvements for Multi-criteria recommender systems. *Ec*, *1*(212), 674–689. https://doi.org/10.1145/2229012.2229065

Konstan, J. A., & Riedl, J. (2012). recommender systems : from algorithms to user experience, 101–123. https://doi.org/10.1007/s11257-011-9112-x

Koren, Y., Ave, P., Park, F., Management, H. D., & Applications, D. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 426–434. https://doi.org/10.1145/1401890.1401944

Melanie Mitchell. (2004). *An introduction to Genetic algorithm*. *Prentice-Hall of India Private Limited, New Delhi*.

Nilashi, M., Bin Ibrahim, O., Ithnin, N., & Sarmin, N. H. (2015). A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA-ANFIS. *Electronic Commerce Research and Applications*, *14*(6), 542–562. https://doi.org/10.1016/j.elerap.2015.08.004

Nilashi, M., Ibrahim, O., & Ithnin, N. (2014). Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and Neuro-Fuzzy system. *KNOWLEDGE-BASED SYSTEMS*. https://doi.org/10.1016/j.knosys.2014.01.006

Reeves, C. R. (1997). Genetic algorithms for the operations researcher. *INFORMS Journal on Computing*, *9*(3), 231–250. https://doi.org/10.1287/ijoc.9.3.231

Ricci, F., Rokach, L., & Shapira, B. (2015). *recommender systems Handbook*. *Springer-Verlag* (Vol. 54). https://doi.org/10.1007/978-0-387-85820-3

Schmitt, L. M. (2001). Theory of genetic algorithms. *Theoretical Computer Science*.

https://doi.org/10.1016/S0304-3975(00)00406-0

Schröder, G., Thiele, M., & Lehner, W. (2011). Setting goals and choosing metrics for recommender system evaluations. *CEUR Workshop Proceedings*, *811*, 78–85.

Shambour, Q., Hourani, M., & Fraihat, S. (2016). An Item-based Multi-Criteria Collaborative Filtering Algorithm for Personalized recommender systems, *7*(8), 274–279.

Sohrabi, B., Toloo, M., Moeini, A., & Nalchigar, S. (2015). Evaluation of recommender systems : A multi-criteria decision making approach. *Iranian Journal of Management Studies*, *8*(4), 589–605.

Toloo, M. (2012). On finding the most BCC-efficient DMU: A new integrated MIP-DEA model. *Applied Mathematical Modelling*, *36*(11), 5515–5520. https://doi.org/10.1016/j.apm.2011.11.074