MULTI-CRITERIA RECOMMENDER SYSTEMS

Dissertation submitted to Jawaharlal Nehru University
in partial fulfilment of the requirement
for the award of the degree of

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND TECHNOLOGY

By

SUSHMA HANS

UNDER THE SUPERVISION OF
PROF. K.K. BHARADWAJ

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-110067
JULY-2008

जवाहरलाल नेहरू विश्वविद्यालय

SCHOOL OF COMPUTER & SYSTEM SCIENCE
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI – 110067 (INDIA)

CERTIFICATE

This is to certify that the dissertation entitled **"Multi-Criteria Recommender Systems"**, being submitted by Miss Sushma to the **School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi** in partial fulfillment of the requirement for the award of the degree of **Master of Technology in Computer Science and Technology**, is a record of original work done by her under the supervision of Prof. K.K. Bharadwaj.

The results reported in this dissertation have not been submitted in part or full to any other University or Institution for the award of any degree or diploma.

Sushma Hans
(Student)

14-07-09

Prof. K. K. Bharadwaj
(Supervisor)

24/7

Prof. Parimala N.
Dean, SC & SS
Jawaharlal Nehru University
New Delhi- 110067
India

Acknowledgement

It gives me immense pleasure to record my humble gratitude towards all those who helped and encouraged me through my dissertation , several of them deserve special mention:

Above all, I wish to express my sincere appreciation to my supervisor **Prof. K.K. Bharadwaj** , School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi for his valuable guidance, enthusiasm & encouragement. He has been an inspiration to me and a great teacher. It was really an unforgettable experience to work under his guidance. I would like to express gratefulness to him for developing in me independent problem solving approach. I would like to appreciate his interest in my queries and their practical suggestions. It would have been really impossible to complete this dissertation without his invaluable support and patience.

I also wish to express a sincere thankfulness to Prof. Parimala N. , Dean and Professor of School of Computer and System Sciences, Jawaharlal Nehru University for providing me this wonderful opportunity and necessary infrastructure to carry out this dissertation. I express my heart-felt indebtedness to my friends and appreciate their help for collecting data for my dissertation. I wish toe express my admiration to all of them for their love and affection and being with me through thick and thin.

At last, I would like to thank my family, especially my niece Jahnvi for their love and support.

Sushma hans

Abstract

The tremendous growth of information on the Internet has been above our abilities to process. Recommender System (RS) has been introduced to help users overcome the information overload problem as it filters out useful information and generate recommendations. A major approach used in recommender systems is Collaborative filtering. Collaborative filtering predicts items which a particular user prefers by using a database about past preferences of users with similar interests. Most commonly used recommendation techniques are based on single rating collaborative filtering. However, an item can be evaluated in many different aspects and in the recent past, multi-criteria rating systems are being explored as an attempt towards producing recommendations with improved accuracy.

In this work we have proposed a framework for incorporating multi-criteria ratings into RS. Firstly we have developed two techniques to compute criteria weights based on Analytical Hierarchical Process (AHP) and Genetic Algorithm (GA). The experimental results clearly show that GA approach gives more accurate weights for various criteria than AHP. Using these techniques for the computation of criteria weights multi-criteria rating collaborative recommendation technique is presented. The experimental results show that proposed multi-criteria RS outperforms the traditional single criteria collaborative filtering recommender system on accuracy of predicted ratings .

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Recommender systems have gained wide-spread acceptance and attracted public interest, leveling the ground for new sales opportunities in e-commerce. These systems constitute an important component of many e-commerce applications by allowing companies to develop long-lasting relationships with the customers [Adomavicius et al., 2001]. These personalization technologies were introduced as a computer-based intelligent technique to help online consumers deal with information overload by signifying which information is most relevant to them. The interest in recommender systems has dramatically increased due to its challenging open issues and it has abundance of useful and practical applications regarding online content and services. The techniques on improving the effectiveness of recommendation system also constitute a problem-rich research area [G. Adomavicius et al., 2005; Lee et al., 2007].

1.1 Recommender Systems

We face far more choices than we can try in the world: which movie is worth watching, where shall I have dinner tonight, which book shall I read, etc. Recommender systems (RSs) have been developed as a solution to the well documented information overload problem [Resnick et al., 1994]. These are designed as a means of sorting through potentially relevant information and making recommendations personalized to the individual user. These internet-based software tools help consumers seek their way through complex online shopping and entertainment sites.

There are three main steps for a typical RS :

I. The user provides some form of input to the system. These inputs can be both implicit and explicit [Resnick et al., 1997]. Ratings provided by users are among explicit inputs whereas time spent reading a website and URLs visited by a user are among implicit inputs.

II. These inputs are used to form a representation of the user's likes and dislikes.

III. The system computes recommendations using these "user profiles".

Some examples of recommender systems are

- *Amazon* – for selecting an interesting book,
- *EBay* – the world's online marketplace,
- *Car Navigator* – for selecting a new car,
- *RentMe* – for finding an apartment,
- *Entree* – for selecting a restaurant and
- *Video Navigator* and *PickAFlick* – for choosing a rental video [R. Burke et al., 1997].

The recommendation mechanisms are usually based on content-based filtering, collaborative filtering and combination of these two systems.

1.2 Taxonomy of Recommender Systems

A large number of recommendation techniques have been developed to date. These are, however, based mainly on content-based and collaborative filtering approach [Wei et al, 2005]. The classification of various techniques of recommender systems is based on the sources of data on which recommendation is based and the use to which that data is put [Burke , 2002]. On this basis, we can distinguish five different recommendation techniques as follows:

1.2.1 Collaborative Filtering

Collaborative filtering (CF) is an approach to making recommendations by finding correlations among users of a recommender system. For CF , the recommendation process is a social activity- collaborative filtering tries to automate word-of-mouth recommendations:. Items are recommended to a new user based on the stated preferences of other, similar users [Manouselis et al., 2007].

CF recommendations consist of three steps:

I. In first step, users provide ratings for the items they have experienced before.

II. Second, the active user is matched with other users in the system. To do so, similarity between active user and other users is computed through correlation coefficients.

III. In the last step, predictions for the items that the active user has not yet rated , but the neighbor has rated are computed [Herlocker et al., 1999; Resnick et al., 1994].

More formally, The rating *r(c,s)* of item s for user c is estimated based on the ratings $r(c_j,s)$ assigned to item s by those users $c_j \in C$ who are "similar" to user c. According to Gediminas Adomavicius [Adomavicius et al., 2005] , various algorithms for CF Recommendations can be grouped into two general classes given below :

Memory-Based Methods

Memory-based algorithms are heuristics that make predictions based on the entire collection of previously rated items by the users. Memory-based methods continuously analyze all user or item data to calculate recommendations [Drachsler et al., 2007]. These systems employ statistical techniques to find set of similar users, known as neighbors. Once a neighborhood of users is formed, these systems use different algorithms to combine the preferences of neighbors to produce a prediction. Two major approaches for similarity computation are Pearson correlation coefficient and cosine-based approach. Both these approaches compute similarity between two users only on the items that both users have rated. Many performance-improving modifications, such as default voting, weighted-majority prediction, inverse user frequency have been proposed as extensions to these standard correlation–based and cosine-based techniques [Adomavicius et al., 2005] .

Model-Based Methods

Model-based algorithms [Breese et al. 1998] use the collection of ratings to learn a model, which is then used to make rating predictions. These algorithms take a probabilistic approach and envision the collaborative filtering process as computing the expected value of a user prediction, with the given user's rating on others items. Various machine learning algorithms such as Bayesian network, Clustering, and Rule-based approaches are used for building models. Bayesian network model formulates a probabilistic model for collaborative filtering approach.

Comparison of Memory-based and Model-based Techniques

- Model-based techniques generalize into a model from the training instances during training and the model need to be updated regularly. The model is then used to make predictions. In contrast, memory-based methods store training instances during training and then retrieved while making predictions.
- Model-based methods learn slowly but make fast predictions whereas memory-based methods learn fast but make slow predictions.

1.2.2 Content-Based Filtering

Content-based approaches to recommendation making are deeply rooted in information retrieval and information retrieval approach [Baudisch, 2001].Content-based filtering (CBF) uses feature of items the user liked in the past to infer new recommendations. For example, Syskill recommends web documents based on users' binary ratings of their interests, Fab system recommends web pages to users [Pazzani et al., 1996; Balabanovic et al., 1997]. CBF systems recommend items based on items' content rather than other users' ratings.

There are essentially four steps for CBF recommendation [Mooney et al., 2000]:

I. The first step is to assemble content data about the items.

II. The second step is to ask user to provide some ratings for the items either randomly given, or can search and find any item he or she likes.

III. The third step is to compile profile of the user using the content information extracted in the first step and rating provided in the second step. Term-frequency/inverse-document frequency weighting [Lang., 1995] and Bayesian learning algorithm [Mooney et al., 2000] are some techniques for profile compilation.

IV. The last step is to match unrated items' content with the user profile obtained from third step and assigning values to the items depending on quality of the match.

The improvement over traditional CBF approaches comes from the use of user profiles that contain information about users' tastes, preferences, and needs.

1.2.3 Knowledge-Based Recommender Systems

Knowledge-based recommender systems suggest items based on logical inferences about user preferences. They use prior knowledge on how recommended items meet the users' needs, and can therefore reason about the relationship between a need and a possible recommendation. The user profile can be any knowledge structure that supports the inference, or just a query formulated by the user, or detailed representation of users' needs.

The restaurant recommender Entree System and several other recent systems employ techniques from case-based reasoning for knowledge-based recommendation. The knowledge used by a knowledge–based recommender can take many forms. Entree system use knowledge of cuisines to infer similarity between restaurants [Burke, 2000].

1.2.4 Demographic-Based Recommender Systems

Demographic-Based recommender systems use prior knowledge on demographic information like age, sex, gender etc about the users and their opinions for the recommended items as basis for recommendations. Demographic information can be used to identify the type of users that like a certain object [Pazzani, 1999]. Demographic techniques form 'people-to-people' correlations like collaborative ones but use demographic data. A stereotype-based Grundy system uses this technique to recommend books based on personal information gathered through interactive dialogue. There are varieties of ways to represent demographic information in a user model. These features could be hand-crafted or extracted from users' home pages [Burke, 2002].

1.2.5 Utility-Based Recommender Systems

Utility-Based recommender systems make suggestions based on a computation of the utility of each item for the user for whom a utility function has to be stored [Manouselis et al., 2007]. Each Utility-Based system like Tête-a-Tête or e-commercial site personal logic (www.personalogic.aol.com/go/gradschools) has different technique for arriving at a user–specific utility function. This utility function works as user profile and system set some constraints to find the best match [Burke, 2002].

1.3 Strengths and weaknesses of Various Recommendation Techniques

A large number of recommendation techniques have been developed as discussed above. All these recommendation techniques have advantages and disadvantages over each other.

Collaborative filtering systems face three problems [Herlocker et al., 1999]:

- *Cold Start Problem* : At the initial use of the system, there are not sufficient users to match with; therefore system cannot be useful for particular user in the beginning phase.

- *Sparsity :* The users-items- ratings matrix is usually very sparse as most users do not rate most items. Also recommender systems are used in domains where there are many items to choose from. Thus, it might be difficult to find highly correlated users.
- *First Rater Problem:* When an item is added to the system, it cannot be recommended to any user unless a user has rated it before.

Content-Based systems also suffer from three problems [Adomavicius et al., 2005] :

- *Limited Content Analysis*: For some domains, there is no content information available, or the content is hard to analyze.
- *Overspecialization:* These systems can suggest only items whose content match with the user's profile. If the user has tastes that are not represented in his/her profile, that item can never be recommended by the system.
- *The Cold Start problem* : Content-based systems also suffer with this problem but only when the active user is at his/her initial stage of use of system.
- Also formulating taste and quality about some new item is not an easy task.

Strengths of Content-Based systems:

- In content-based systems, active user does not have to wait for other users to use the system and provide ratings for items in order to receive good recommendations.
- *First rater problem* and *sparsity problems* are also not an issue for Content-based systems, because Content –based systems do not try to match with other users.

Strengths of Collaborative Filtering systems:

- Collaborative filtering systems do not suffer with *limited content analysis* problem as these systems do not look for content. This technique is based on similar users.
- Also the problem of *overspecialization* occurred in Content-based systems is not an issue for these systems because the neighbors might have tastes that the active

user's profile do not represent. This strength is also referred as *'outside the box'* recommendation ability [Mooney et al. 2002].

- The problem regarding quality and taste of new items is also partially solved by CF systems because taste and quality is entered by the users as ratings. Any item regardless of content can be recommended by these systems. This also means CF can recommend things from different genres.

The benefit of demographic recommender systems is that it may not require a history of user ratings as required in collaborative and content-based techniques. So these systems do not suffer with *cold start problem.* Still there are not many recommender systems using demographic data as it has the problem of gathering the requisite demographic information: those users are hesitant to disclose.

The Strength of utility-based recommendation is that it can factor non-product attributes into the utility computation. These systems are also free from cold start and sparsity problems as its recommendations are not based on accumulated statistical evidence. This approach has more beneficial than other approaches as it can incorporate many different factors for recommendations. But the central problem for this technique is to create the utility function for each user.

The benefit of using a knowledge–based system is that there is no bootstrapping problem as the recommendations are based on prior knowledge; no learning time is required before making good recommendations. Knowledge–based approaches have functional knowledge about how a particular item meets a particular user need, and can therefore reason about what product meets the user's requirements. But this is also a drawback since pure knowledge-based systems can only make pre-coded recommendations and not able to adapt to the changing domains or individual user.

1.4 Hybrid Recommender Systems

Hybrid recommender systems combine two or more recommender algorithms. The reason is to make use of their combined strengths and to level out their corresponding weaknesses [Burke, 2002]. Most of the hybrid approaches are geared towards unifying collaborative filtering with content-based filtering under one single framework, leveraging synergetic effects and mitigating inherent deficiencies of either paradigm. A system called Learning Intelligent Book Recommending Agent (LIBRA), employs collaborative filtering along with content based recommendation.

Burke lists an entire plethora of hybridization methods to compare of recommendation algorithms. Table 1.1 shows some combination methods such as Weighted , Switching , Mixed, Feature Combination , Cascade , Feature Augmentation , Meta-Level .

Hybridization method	Description
Weighted	The scores of several recommendation techniques are combined together to produce a single recommendation.
Switching	The system switches between recommendation techniques depending on current situation.
Mixed	Recommendations from several different recommenders are presented at the same time.
Feature combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm.
Cascade	One recommendation refines the recommendation given by another.
Feature augmentation	Output from one technique is used as input feature to another.
Meta-level	The model learned by one recommender is used as input to another.

Table 1.1 : Hybridization methods [Burke , 2002]

Some examples of Hybrid RSs are:

- P-Tango system [Claypool et al., 1999] uses weighted combination of collaborative and content-based recommendation.
- The DailyLearner system is based on hybridization of content/collaborative recommenders. If one recommender fails to make good recommendation, then it switches to other recommender.
- The PTV system [Smyth et al., 2000] uses mixed hybrid to amass a recommended program of television viewing.
- Inductive rule learner [Basu et al.,1998] uses both user ratings and content features to recommend movies.
- The restaurant recommender EntreeC, first uses knowledge of restaurants to make recommendations based on user's interests. It uses knowledge-based system at first level and Collaborative filtering system at second level [Burke, 2002].
- Amazon.com [Linden et al., 2003; Mooney et al., 1999], GroupLens research team also used feature augmentation hybrid.
- Fab, the web filtering system, was the first meta-level hybrid system.

1.5 Multicriteria Recommender Systems

Recommendations based on multicriteria ratings emerged as challenging and largely unexplored issue in the field of recommender systems. Multicriteria ratings provide additional information about user preferences for different aspects or components of an item. As an example , for movie RS , in addition to the overall rating, Gediminas Adomavicius and YoungOk Kwon [Adomavicius et al., 2007] used four rating criteria that are story, acting, direction, and visuals. First take the single rating systems where user provides single rating for each movie they have seen as presented in Figure 1.1:

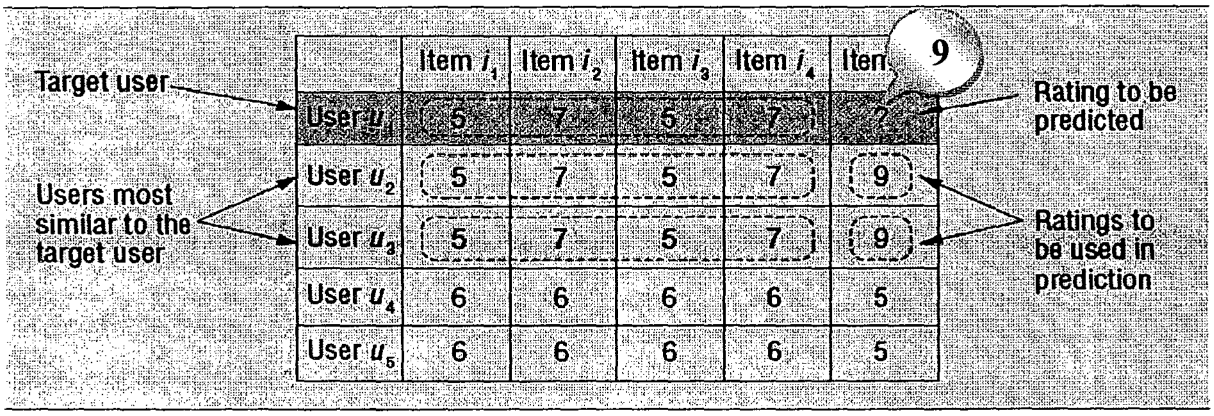


Figure 1.1: Collaborative filtering in a single-rating setting [Adomavicius et al., 2007].

So traditional user-based collaborative–filtering approach would estimate any rating that user u would give to yet unseen movie i according to how user u' who are similar to active user u rated movie i. So rating prediction shall be more accurate if system determines accurately the true peers of active user. Two dimensional collaborative filtering calculates similarity between two users on basis of how their ratings are for the movies they both have seen [Adomavicius et al. , 2007].

Figure 1.1 show this estimation process with a simple example. Suppose we have five users u_1 ,...., u_5 and five movies i_1,....,i_5. and assume that the recommender system needs to estimate how much the active user u_1 would like movie i_5. The traditional collaborative-filtering approach finds the users that are closest to u_1 who have seen movie i_5. In this case, we can figure out that user u_2 and u_3 are perfect matches for user u_1 because their ratings for common movies are exactly the same. User u_2 and u_3 rate movie i_5 as 9, the system will result the value of target rating $R(u_1,i_5)$ as 9.

Now we take the whole scenario in multicriteria setting. In this setting, we assume that the system also ask the user to provide the feedback about movie on four specific criteria- story, acting , direction, visuals and overall rating in this case is simple average of the four individual criteria ratings as shown in Figure 1.2. This additional information helps us to figure out that user u_2 and u_3 are very much different in their tastes and preferences from active user u_1, even though their overall ratings are perfectly same. User u_4 and u_5 are much better matches for user u_1 in this example.

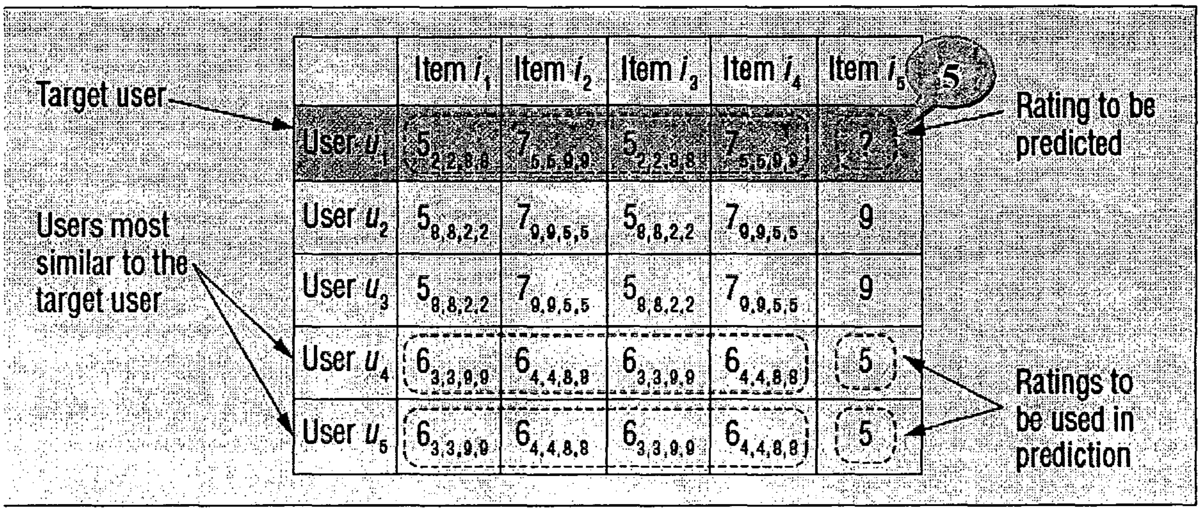


Figure 1.2: Collaborative filtering in a multicriteria setting, where the overall rating for each movie *I* is a simple average of four rating criteria [Adomavicius et al., 2005] : story, acting, direction, and visuals.

So the system will predict a value of 5 for the target rating $R(u_1,i_5)$ as both users u_4 and u_5 rate movie i_5 as 5.

Therefore, recommendation accuracy can be improved by incorporating multicriteria ratings in recommendation algorithms. Each user has different preferences, so importance of each criterion in multi-criteria system varies from user to user. These preferences can be represented in the form of weights for various criteria. We have explored Analytical Hierarchical Process (AHP) to compute these weights. Alternatively we have employed Genetic Algorithm to learn these weights for each user.

Analytical Hierarchical Process

Analytical hierarchical process (AHP) allows decision makers to model a complex problem into a hierarchical structure showing the relationship of the goal, objectives (criteria), sub-objectives and alternatives. Analytical hierarchical process enables decision makers to derive ratio scale priorities or weights as opposed to arbitrarily assigning them [Forman et al., 2001].

Genetic Algorithms

John Holland [Holland , 1975] introduced a method of studying natural adaptive system and designing artificial adaptive systems based on Darwinian natural selection and Mendelian genetics. This method eliminates weak elements by favoring retention of optimal and near optimal individuals (survival of the fittest), and recombines features of good individuals to perhaps make better individuals. Genetic algorithms (GAs) use this method to search the representation space of artificial adaptive systems, that represent a problem's search space as sequences (strings) of symbols chosen from some alphabet (usually a binary alphabet). The algorithm performs optimization by manipulating a finite population of chromosomes. In each of a number of cycles called generations, the GA creates a set of new chromosomes by crossover, inversion, and mutation, which correlate to processes in natural production.

1.6 Thesis Roadmap

The rest of the thesis is organized as follows. Chapter 2 presents relevant background information on Multicriteria recommender systems, Genetic Algorithm technique and analytical hierarchical process . The proposed Multicriteria Recommender System is discussed in chapter 3. Experimental results are presented in chapter 4. Finally, chapter 5 presents conclusions and future work.

Chapter 2

BACKGROUND AND LITERATURE REVIEW

In this chapter, we focus on extending the concept of single criterion ratings to multi-criteria ones, i.e., an item can be evaluated in many different aspects. Different weights for all criteria have to be provided to reflect the relative importance while making decisions among all the alternatives. We have also described two techniques, Genetic Algorithms and Analytical Hierarchical Process, that are used to calculate weights of different criteria [Lee et al. 2007].

2.1 Multicriteria Recommender Systems

As recommendation systems emerge as an independent research area, the rating structure plays a critical role in recommendation process. Particularly where recommendations are based on the opinion of others, it is crucial to take into consideration the multiple criteria that affect the users' opinions in order to make more effective recommendations [Manouselis et al., 2007]. Most of the current recommendation systems deal with single criterion ratings, such as ratings of books etc. Nevertheless, in some applications, such as movie recommender systems and restaurant recommendation systems, it is vital to incorporate multi-criteria ratings into recommendation methods to increase recommendation accuracy [Adomavicius et al., 2005]. This extension of multi-criteria ratings has significant impacts on existing recommendation techniques. Multicriteria recommender systems intend to find items that are most useful to each user like single-criteria rating systems. So, the system must be able to predict each item's overall rating for each user so that items can be compared on the basis of their overall ratings and recommend the best items to the users [Li et al., 2008].

In single-criteria rating systems, the extent to which user is interested in an item can be represented in 2D *User* × *Item* space. In multi-criteria framework, a 3-order tensor is used to represent 3D *User* × *Item* × *Criterion* data. More formally, the general form of rating function in multicriteria recommender system is

$$R: Users \times items \rightarrow R_0 \times R_1 \times \ldots\ldots\ldots \times R_k$$

where R_0 is the set of possible overall rating values, and R_i represents the possible rating values for each individual criterion i $(i = 1,\ldots,k)$ on numerical scale [Adomavicius et al., 2007].

2.1.1 Incorporating Multi-Attribute Utility Theory in Collaborative Filtering Approach

Multi-Attribute Utility Theory is a decision making method used when decision maker has to take several competing objective into account. It is a normative/prescriptive method because it tells what we should do [Schmitt et al., 2002].

Engaging Multi-Attribute Utility Theory (MAUT) in Collaborative filtering approach [Manouselis et al., 2006a ; Manouselis et al., 2006b] , the recommendation problem can be defined as a decision problem with multiple variables, called multi-*attribute collaborative filtering* in the following way. The multiple attributes describing an item s are defined as a set of criteria which are used to evaluate the item. In this theory, the utility function $U^c(s)$ is referred to as *total utility* of an item *s*, which can be calculated by synthesizing the *partial utilities* of item *s* on each one of the criteria.

Assume that there is no uncertainty during decision making , the total utility of a item $s \in S$ for a user $c \in C$ can be expressed as:

$$U^c(s) = \sum_{i=1}^{n} u_i^c(s) = \sum_{i=1}^{n} w_i^c\, g_i^c(s)$$

where $u_i^s(s)$ is the partial utility function of the item s on criterion g_i for the user c, $g_i^c(s)$ is the evaluation that user c has given to item s on criterion g_i , and w_i^c is the weight indicating the importance of criterion g_i for the particular user c, with:

$$\sum_{i=1}^{n} w_i^c = 1$$

Taking full advantage of multicriteria ratings in recommender systems, Gediminas Adomavicius and YoungOk Kwon presents two new approaches i.e. Similarity-based approach and Aggregation-function-based approach that involves multicriteria ratings [Felfernig et al., 2007]. Different weights have to be provided for all criteria to reflect the relative importance when making decisions among all the alternatives. The major difficulty is that the weights of diverse criteria vary with different user and vary with time. For example, some people give price as topmost priority when choosing restaurants, while other may not think so. Consequently, a proper recommendation technique should not make an assumption that the weights of all criteria are time-variant and exactly known. We have used two popular techniques to calculate weights for all criteria [Lee et al. , 2007]:

- Analytical Hierarchical Process
- Genetic Algorithms

Until recently, the most common aggregation tool used in multi-criteria decision-making is the weighted arithmetic mean, movies are then ranked according to the aggregated scores they have been assigned [Plantie et al.,1996].

2.2 Introduction of Analytical Hierarchical Process

"Analytical Hierarchy Process (AHP) is an approach to decision making that involves structuring multiple choice criteria into a hierarchy, assessing the relative importance of these criteria, comparing alternatives for each criterion, and determining an overall ranking of the alternatives",

The concept of AHP was developed, amongst other theories, by **Thomas Saaty**, an American mathematician working at the University of Pittsburgh. AHP is a comprehensive, logical and structured framework that allows to improve understanding of complex decisions by decomposing the problem. This method has been found to be an effective and practical approach for complex and unstructured decisions. The decision-maker judges the importance of each criterion in pair-wise comparisons. The result of AHP is a prioritized ranking or weighting of each decision alternative [haq et al., 2005]. Basically there are three steps for considering decision problems by AHP : Constructing hierarchies, comparative judgement , and synthesis of priorities, described as follows:

❖ Establishment of a structural hierarchy

The AHP begins with the development of decision hierarchy with an objective, alternatives and criteria. The objective or the overall goal of the decision is represented at the top level of the hierarchy. The criteria and sub-criteria contributing to the decision form intermediate levels. Finally, the decision alternatives or selection choices are represented at the last level of the hierarchy (Figure 2.1).

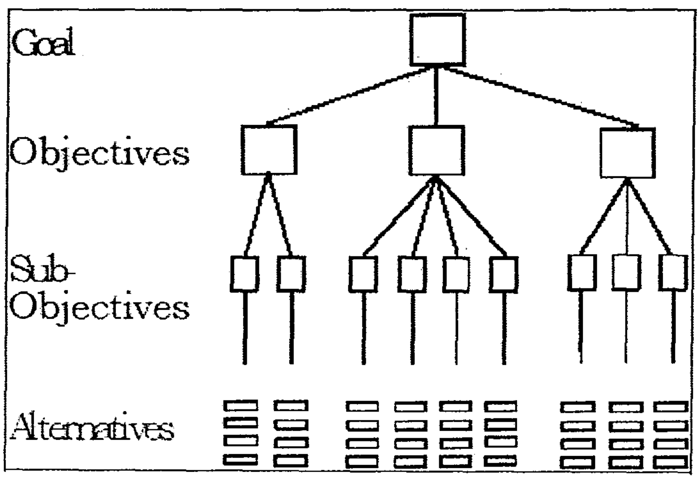


Figure 2.1: Decision Hierarchy

❖ **Establishment of comparative judgements**

Once the hierarchy has been structured, next the decision-makers individually express their opinions regarding the relative importance of the criteria and preferences among the alternatives using pair-wise comparisons. The pair-wise comparisons generate a matrix of relative rankings for each level of the hierarchy. The number of matrices depends on the number elements at each level. The order of the matrix depends on number of elements at the lower level that it links to.

❖ **Synthesis of priorities and the measurement of consistency**

After all matrices are developed and pair-wise comparisons are obtained, these preferences undergo a synthesis process to calculate priority weight vector for the criteria, global weights, and maximum eigenvalue (λ_{max}) for each matrix. Also check the consistency property matrices to ensure that the judgements of decision makers are consistent [Steiguer et al., 2003 ; Haq et al., 2006].

By aggregating the relative weights of decision elements, we can obtain an overall rating for the alternatives.

2.2.1 Why AHP uses Eigenvalues and Eigenvectors

Assume we already knew the weights of all criteria, then we can express them in pair-wise comparison matrix as shown:

$$\underline{A} = \begin{bmatrix} w_1/w_1 & w_1/w_2 & w_1/w_3 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & w_2/w_3 & \cdots & w_2/w_n \\ w_3/w_1 & w_3/w_2 & w_3/w_3 & \cdots & w_3/w_n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_n/w_1 & w_n/w_2 & w_n/w_3 & \cdots & w_n/w_n \end{bmatrix}$$

To find out the vector of weights given these ratios, we can take the matrix product of the matrix A with vector w [Forman et al., 2001]:

$$\begin{bmatrix} w_1/w_1 & w_1/w_2 & w_1/w_3 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & w_2/w_3 & \cdots & w_2/w_n \\ w_3/w_1 & w_3/w_2 & w_3/w_3 & \cdots & w_3/w_n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_n/w_1 & w_n/w_2 & w_n/w_3 & \cdots & w_n/w_n \end{bmatrix} * \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \cdots \\ \cdots \\ \cdots \\ w_n \end{bmatrix} = \begin{bmatrix} nw_1 \\ nw_2 \\ nw_3 \\ \cdots \\ \cdots \\ \cdots \\ nw_n \end{bmatrix}$$

$$\underline{A} \qquad * \ \underline{w} \ = n\,\underline{w}$$

If we knew A, but not w ,we could solve the above for w. The problem of solving for a nonzero solution to this set of equations is known as an eigenvalue problem:

$$\mathbf{A\, w = \lambda\, w}$$

Notice that each column of $\underline{A}$ is a constant multiple of $\underline{w}$. Thus, $\underline{w}$ can be obtained by normalizing any column of $\underline{A}$.

The matrix $\underline{A}$ is said to be strongly consistent if

$a_{ik}a_{kj} = a_{ij}$ for all i, j.

For the inconsistent case, the eigenvalues problem can be solved as:

$$\mathbf{A\, w} = \lambda_{max}\, \mathbf{w}$$

2.2.2 Strengths and Drawbacks of AHP

The strengths of using AHP are

❖ **Flexible tool :** This technique is very flexible as it make decisions through multiple and conflicting criteria. It also takes care of qualitative and quantitative aspects of a decision.

❖ **Robust :** AHP is considered as robust as it captures both subjective and objective aspects of a decision. It also keeps a check on consistency of the decision maker's evaluations.

❖ **Easy to implement :** This technique require only a few matrix manipulations. So it is very easy to implement AHP for weight calculations.

There are few drawbacks also associated with this technique-

❖ Every evaluation in AHP require to express how well two criteria are compared to each other. This type of comparative data is difficult to obtain. Also the number of pair-wise comparisons grows quadratically with the number of criteria and scenarios.

2.2.3 Example of Application

Lets take the decision of buying a best car and there are three alternatives Ford Taurus, Lexus and Saab 9000. The criteria and sub criteria for choosing a car are represented in a hierarchy as shown in Figure 2.2.

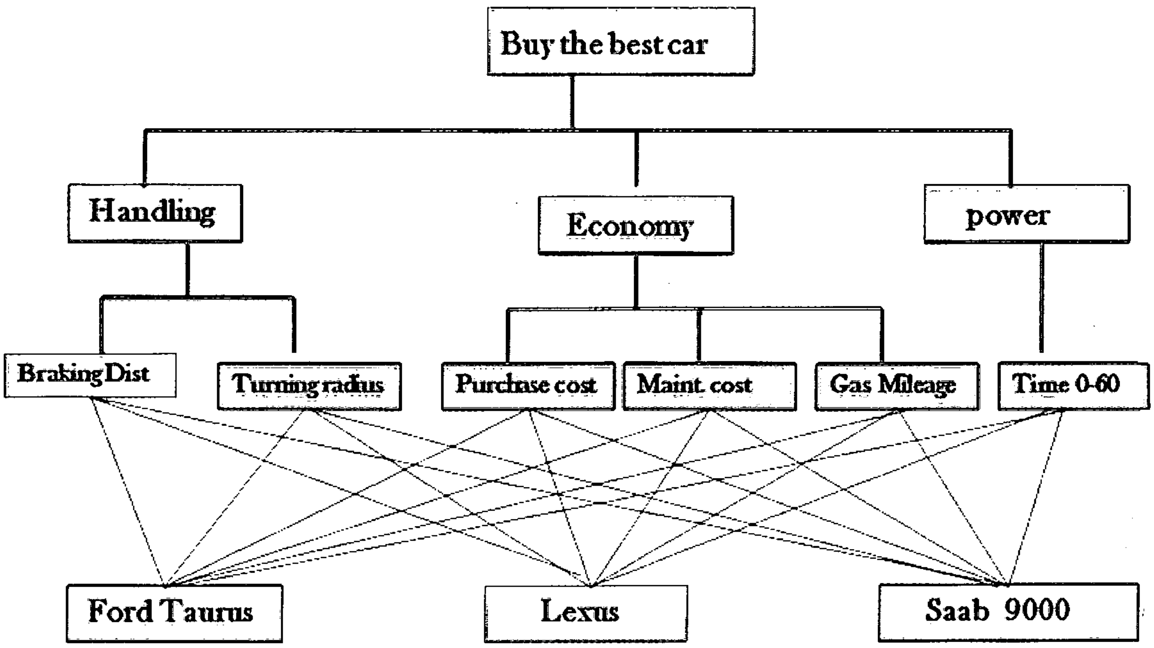


Figure 2.2: Example of buying a car

Pairwise Comparison

Pairwise comparisons are fundamental to AHP. To build the judgement matrix, two types of comparisons are made: which criterion is more important and how much it is important than other criteria. The following table show the numerical ratings recommended for verbal preferences. We ask the user to state preference between cars considered two at a time (Pairwise) starting from lowest sub criteria. For example, purchase cost (P) and Maintenance cost (M), P and gas Mileage (G), M and G. A 3-3 matrix is formed from these comparisons. In the same way, other sub-criteria under the same criteria are compared and same process will go on until we achieve the goal.

The Table 2.1 shows the numerical ratings recommended for the verbal preferences expressed by the decision –maker. This is generally used scale.

Verbal Preference	Numerical Rating
Extremely preferred	9
very strongly to extremely	8
Very strongly	7
Strongly to very strongly	6
Strongly	5
Moderately to strongly	4
Moderately	3
Equally to moderately	2
Equally	1

Table 2.1: 1-9 Scale

Suppose we get the following matrix:

	P	M	G
P	1	3	5
M	1/3	1	3
G	1/5	1/3	1

Ratings mean that P is 3 times more important than M and P is 5 times more important than G. What's wrong with this matrix?

The ratings are inconsistent

Ratings should be consistent in two ways:

(1) Ratings should be transitive:– That means that If A is better than B and B is better than C then A must be better than C

(2) *Ratings should be numerically consistent* :– In car example we made 1 more comparison than we needed We know that P = 3M and P = 5G

$\Longrightarrow$ 3M=5G $\Longrightarrow$ M=5/3G

So consistent matrix for the car example would look like:

	P	M	G
P	1	3	5
M	1/3	1	5/3
G	1/5	3/5	1

Note that matrix has Rank = 1, that means that all rows are multiples of each other. It is easy to compute weights of this matrix. As rows are multiples of each other, we can compute weights by normalizing any column.

• We get $w_P = \frac{15}{23} = 0.65$, $w_M = \frac{5}{23} = 0.22$, $w_G = \frac{3}{23} = 0.13$

Weights for inconsistent matrix is obtained by computing for the inconsistent matrix

	P	M	G
P	1	3	5
M	1/3	1	3
G	1/5	1/3	1

$= A$

We need to solve the equation : $Aw = \lambda_{max} w$, where w is vector of weights.

By solving $\det(A - \lambda I) = 0$ We get: $\lambda_{max} = 3.039$. Eigen vector or weight vector corresponding to $\lambda_{max} = 3.039$ and normalize are $W_p = 0.64$, $W_m = 0.26$ $W_g = 0.10$.

2.3 Genetic Algorithm Approach

Genetic algorithm (GA) Approach is an artificial intelligence procedure inspired by Darwin's theory of evolution. GA uses the idea of survival of the fittest by progressively accepting better solutions to the problems [Min et al.,2005]. Each individual solution in GA is represented by a string called the chromosome. The key feature of a GA is the manipulation of a population whose individuals are characterized by possessing a chromosome. Each chromosome is a simple coding of a potential solution to the problem domain. With successful generations of population through reproduction and recombination operators such as mutation and crossover, the overall quality of the population as assessed by the fitness function improves [Lim et al., 1996]. Reproduction is an exploitative mechanism. Exploiting is achieved by favoring chromosomes that are more fit so as to get better chances of converging towards an optimal region. Crossover and mutation are two recombination operators used for reproduction and to explore the search space. Crossover is a simple but exploratory tool that is capable of retaining and propagating what is learnt from previous generations. Mutation, on the other hand, offers a mechanism to maintain the population diversity. A good Genetic Algorithm is one that is intelligent enough to strike a balance between exploitation and exploration, achieved by assigning proper values to GA parameters such as population size, probability of crossover and mutation.

The structure for simple GA approach is shown in Figure 2.3

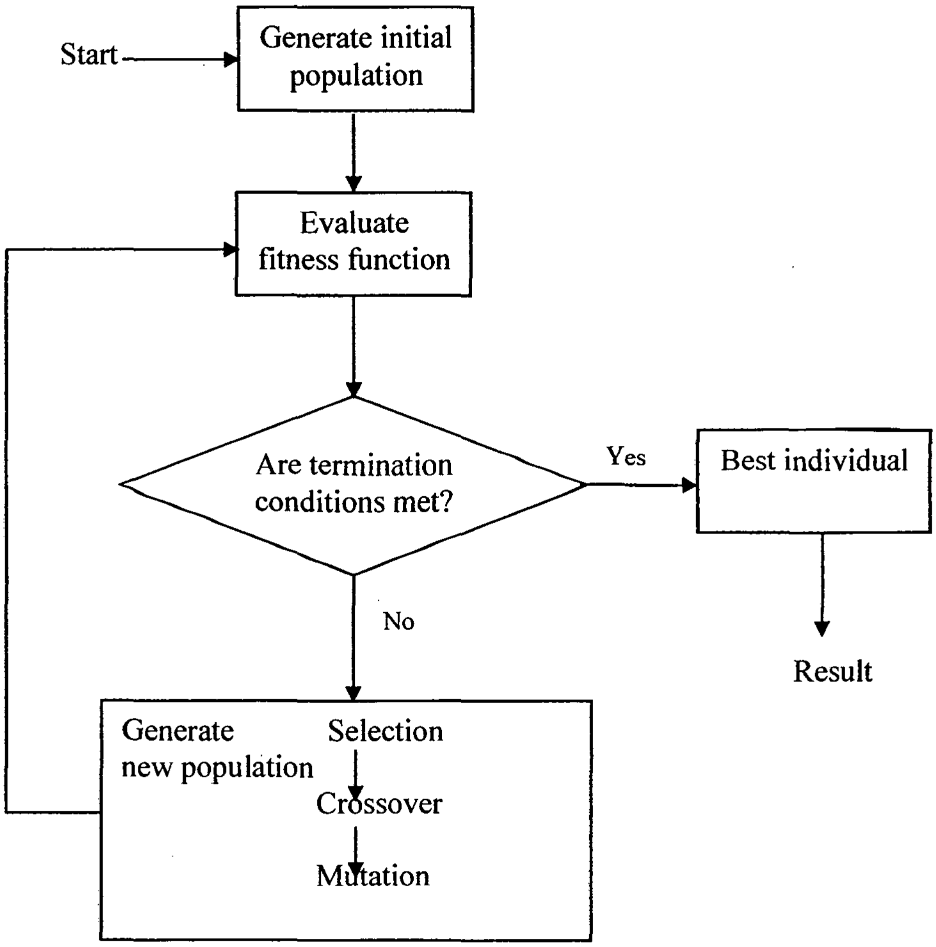


Figure 2.3 Genetic Algorithm

Steps to follow for GA approach are:

1) Create a population of random individuals in which each individual represents a possible solution to the problem at hand.
2) Evaluate each individual's fitness – its ability to solve the specified problem.
3) Select individual population members to be parents.
4) Produce offspring by recombining parent material via crossover and mutation and add them to the population.
5) Evaluate offspring's fitness.
6) Repeat steps (iii) – (v) until a solution with the desired fitness goal is obtained.

2.3.1 Chromosome Representation

In Genetic Algorithms, a chromosome is often represented by a fixed-length of genes, where each gene is a small part of a candidate solution. Chromosome can be represented in a number of ways i.e. alphabets or string of bits, so that each gene can take on either 0 or 1 (Figure 2.4).

1	0	1	1	0	1	1	0	1	1

Figure 2.4: Example of the binary encoding traditionally adopted with the GA.

2.3.2 Fitness Function

The most important concept of Genetic Algorithm is the fitness function. GA is driven by the fitness measure. Fitness function depends on the type of problem and varies greatly according to the problem. The fitness assigns a value to each chromosome in the population. This fitness function actually determines which chromosome will continue to exist in the next generation. The value of this fitness function is the basis for selection strategy so its choice is very vital.

2.3.3 Selection methods

The selection is a two-step process:

- Firstly, calculate the fitness value for each chromosome.
- In the second step, assign a probability for each chromosome in the population.

Selection process allows strings with higher fitness to appear with higher probability in the next generation. There are numerous ways to select a chromosome to be replicated to the population, including fitness proportionate selection, tournament selection, steady state selection, rank based selection, and as an addition to many selection methods an *elitism* approach, which force the GA to keep some number of the best individual at each generation.

2.3.4 Genetic Operators

Genetic operators are one of the most significant components of GA. Genetic operators are applied to individual that are chosen probabilistically from the population on the basis of fitness. Two basic operators that mainly influence the performance are:

- **Crossover operator**

Crossover operation is performed between two selected individuals, called parents, by exchanging parts of their strings, starting from a randomly chosen crossover point. This operator tends to enable to the genetic algorithms to move towards promising regions of the search space. There are several ways of performing crossover. The simplest is the one-point crossover. The two parents involved in crossover are determined via probability of crossover Pc. A certain chromosome is going to be selected for crossover if a random generated number is smaller than Pc. Crossover point is also selected at random between 1 to the length of string and exchange all bits of two parents after crossover point as shown in Figure 2.5.

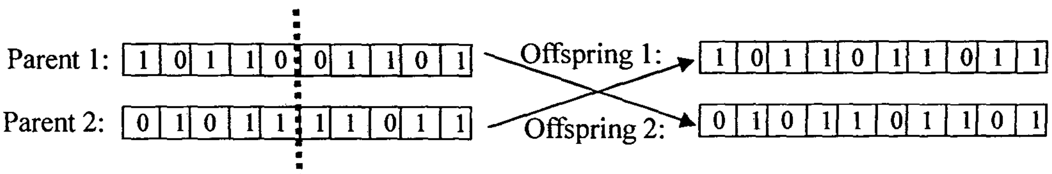


Figure 2.5: Crossover operator

- **Mutation operator**

Mutation operator act on a single individual at a time. Mutation replaces the value of a gene with a random-generated value. Mutation helps to increase the genetic diversity of the population and to avoid the local convergence problem. P_m is the probability of mutation and it tells if a certain chromosome undergoes a mutation. A random number is generated and if it is smaller than P_m, the mutation is going to be applied to the chromosome.

Parent | 0 | 1 | 0 | 1 | 1 | **0** | 1 | 1 | 0 | 1 | Offspring: | 0 | 1 | 0 | 1 | 1 | **1** | 1 | 1 | 0 | 1 |

Figure 2.6: GA Mutation

It is implemented by selecting a random string location and changing its value from 0 to 1 or vice versa, as shown in Figure 2.6.

Chapter 3

A FRAMEWORK FOR MULTICRITERIA RECOMMENDER SYSTEMS

Multicriteria recommender systems have the advantage that they consider more than one criterion that may affect the potential user's decision. Multicriteria rating systems have more information about the users and items to use in recommendation process. So multicriteria ratings for an item can provide us more precise approximations to the similarity between two users since they give a good insight into why users like the item whereas single-criteria rating (overall rating) can only tell us how much user like it [Manouselis et al., 2006a; , Li et al., 2008]. Multicriteria ratings correspond to user preferences for different components of an item, such as story, acting, direction and visuals in the case of movies. So we cannot assume an item's overall rating is independent of other criteria ratings; rather it serves as some aggregation function f of the item's multicriteria ratings [Adomavicius et al., 2007]:

$$r_0 = f(r_1, r_2, \ldots\ldots r_k) \qquad (3.1)$$

In movie recommendation application, as a specific case, we are using aggregation function f as

$$r_0 = W_S\, r_1 + W_A\, r_2 + W_D\, r_3 + W_V\, r_4 \qquad (3.2)$$

where r_0 is the overall rating and r_1, r_2, r_3, r_4 are the ratings for four criteria (story, acting, direction, visuals) with corresponding feature weights W_S, W_A, W_D, W_V. These weights represent the priority that user offer to the criteria while selecting movie. For instance, in this application, story criterion might have a high priority i.e. movies with high story ratings are liked on the whole by some users, regardless of other criteria ratings. So if system finds that movie's story rating is high, it must predict that overall rating will also be high in order to be accurate.

3.1 Feature Weighting

We have used following two feature weighting methods that work on different type of data to compute weights and experiments are conducted to compare these two methods on the data set of hindi movies collected through survey due to non- availability of an appropriate data set .

3.1.1 Eigen Vector method :

Eigen vector method works on comparative data. For example, in movie recommendation application, we have four criteria i.e. story, acting, direction and visuals. We carried out a survey and collected data as shown in Table 3.1:

Story-Acting	Story-Direction	Story-Visuals	Acting-Direction	Acting-Visuals	Direction-Visuals
3:2	3:2	7:3	3:2	7:3	7:3

Table 3.1 : Comparative Data for Eigen Vector Method

This data represent that this particular user prefer story 3/2 times than acting, story 3/2 times than direction, story 7/3 times than visuals and so on.

Main steps to compute feature weights using Eigen vector method :

I. **Matrix Construction:** The above data can be represented in the form of a matrix as

Shown below:

$$A = \begin{array}{c} \\ S \\ A \\ D \\ V \end{array} \begin{array}{c} \begin{array}{cccc} S & A & D & V \end{array} \\ \begin{bmatrix} 1 & 3/2 & 3/2 & 7/3 \\ 2/3 & 1 & 3/2 & 7/3 \\ 2/3 & 2/3 & 1 & 7/3 \\ 3/7 & 3/7 & 3/7 & 1 \end{bmatrix} \end{array}$$

Here

S = Story

A = Acting

D = Direction

V = Visuals

II. **Calculate λ_{max} :** Next step is to calculate maximum eigen value λ_{max} . For this, We need to solve the equation : $Aw = \lambda_{max}w$, by eigen vector method where w is vector of weights.

By solving $\det(A - \lambda I) = 0$ We get: $\lambda_{max} = 4.04$.

III. **Check Consistency:** we need to check the consistency property of matrices to ensure that the judgements are consistent. For consistent judgement we need that consistent ratio must be less than 0.1. To compute the consistency ratio, we follow the following steps:

- Compute Consistency Index (C.I.) = $\dfrac{\lambda_{max} - n}{n - 1}$ (3.3)

From movie example :

C.I. = (4.04 - 4)/ (4-1) = 0.04/ 3 = .013

- Another measure compares C.I. with randomly generated ones (Table 3.2)

C.R. = C.I. / R.I. where R.I. is the random index (3.4)

n	1	2	3	4	5	6	7	8
R.I.	0	0	.52	.89	1.11	1.25	1.35	1.4

Table 3.2: Random Index For Consistency Measure

For movie example:

C.I. = 0.013

n = 4

R.I. = 0.89 (Table 3.2)

So, C.R. = C.I. / R.I. = 0.013 / 0.89 = 0.0146

As C.R < 0.1 therefore this indicates sufficient consistency.

So we need to check each user's matrix for sufficient consistency.

IV. **Calculate Weights:** Weights for all criteria are obtained by calculating eigen vector (weight vector) corresponding to λ_{max} i.e. 4.04 in this particular example.

Eigen vector or weight vector corresponding to and normalize are

$\mathbf{W_S = 0.35}$, $\mathbf{W_A = 0.28}$, $\mathbf{W_D = 0.34}$, $\mathbf{W_V = 0.13}$

In this way we get weights for all the users. But one problem with this technique is that it is difficult to get this type of data. Also sometimes the data is so inconsistent that it does not provide consistent judgement.

3.1.2 Genetic Algorithm Approach

Second technique used to evolve feature weights for active user is Genetic Algorithms (GA). This technique works on user provided multicriteria ratings for different movies. An elitist genetic algorithm approach chosen is chosen for this task where a quarter of the best individuals in the population are kept for the next generation [Ujjin et al., 2004]. The chromosome representation for weights and fitness function used by this technique are described below:

- **Chromosome Representation:** We need to find weights for four criteria in the movie recommendation dataset. A simple unsigned binary genetic encoding is used in this implementation, using 3 bits for each of the four criteria. For example (W_S = 5 , W_A = 2 , W_D = 4 & W_V =3) is encoded as shown in Figure 3.1.

Story (W_S)			Acting (W_A)			Direction (W_D)			Visuals (W_V)		
1	0	1	0	1	0	1	0	0	0	1	1

Figure 3.1 : Chromosome Representation for Weights

The binary encoding is mapped to the decimal values and feature weights are then calculated from these real values. Weighting value for each criterion can be found by dividing the value by the total value. The sum of all the weights will then add up to unity [Ujjin et al., 2004]:

W_S	W_A	W_D	W_V
5	2	4	3

Normalize

W_S	W_A	W_D	W_V
0.36	0.14	0.29	0.21

➢ **Fitness Function:** The choice of a fitness function is usually very specific to the problem under consideration. Calculating the fitness for this application is not trivial. To find the fitness score for the evolved set of weights, first map the binary chromosome is mapped to decimal and normalized to get the set of weights corresponding to four criteria. We predict the overall rating through formula (3.1) i.e.

$$r_0' = W_S r_1 + W_A r_2 + W_D r_3 + W_V r_4$$

$$\text{Fitness} = MAE(u_i) = \frac{1}{|S_i|} \sum_{k=1}^{|S_i|} |pr_{i,k} - r_{i,k}| \qquad (3.5)$$

r_0' is calculated for all rated movies by the user and predict overall rating for each movie. The average of the difference between the actual (r_0) and predicted ratings (r_0') for all the movies in the data set of the given user give the fitness score for that set of weights [Ujjin et al., 2004]. This is a case of minimization problem and therefore low fitness score signify that the weights are more fit for the user.

Steps employed to learn weights for movie recommender system are:

a) Initially a set of 20 chromosomes representing weights for four criteria is randomly generated.

b) The fitness of each chromosome in the population is evaluated using a fitness function described above and all chromosomes are arranged in ascending order of fitness.

c) For each generation,

- A new population is generated by selecting chromosomes by selecting chromosomes pair-wise from top eight chromosomes of the population and Crossover is performed and Mutation is performed on the last two chromosomes at randomly generated points. This gives us 10 new offspring that would replace the last 10 chromosomes having high fitness score.
- Now, fitness of all 20 chromosomes is calculated using formula (3.5) and arranged all in ascending order of fitness.

d) Process is terminated when solution becomes stable and not changing with further iterations. It is observed that after 200 iterations, the solution becomes almost stable for most of the users.

3.2 Collaborative Filtering for Multicriteria Ratings

To explore the affect of multicriteria ratings in recommender systems, we had collected multicriteria ratings for 25 movies from 100 users through a survey . One such sample is shown in Table 3.3:

S. No.	Movie	Overall rating	Story	Acting	Direction	Visuals
1	Sholay	7	8	8	7	5
2	Hum Aapke Hain Kaun	6	7	7	6	5
3	Dilwale Dulhaniya Le Jayege	8	8	8	8	7
4	Raja Hindustani	5	5	6	6	5
5	Dil Toh Pagal Hai	6	5	7	7	6
6	Kuch Kuch Hota Hai	8	7	9	8	7
7	Kaho Na Pyar Hai	7	7	7	8	7

8	Hum Dil De Chuke Sanam	8	7	9	8	6
9	Hera Pheri	9	8	9	9	7
10	Golmaal(old)	8	8	9	8	7
11	Golmaal(New)	8	7	8	7	6
12	Chak De India	8	8	8	8	6
13	Om Shanti Om	6	5	8	8	7
14	Welcome	6	6	7	6	4
15	Taare Zameen Par	9	9	9	9	6
16	Rang De Basanti	8	8	8	7	5
17	Jo Jita Wohi Sikandar	8	8	9	8	6
18	Yuva	7	7	8	7	5
19	Kabhi Khushi Kabhi Gham	8	8	8	9	8
20	Dhoom 1	7	8	7	7	8
21	Dhoom 2	8	9	8	8	8
22	Krish	6	5	7	8	8
23	Koi Mil Gaya	8	8	7	8	8
24	Phir Hera Pheri	3	3	7	4	1
25	Bheja Fry	7	7	7	5	4

Table 3.3 : User 's Multicriteria Ratings sample

To perform comparison between single criteria and multicriteria rating systems, all the movie items that the active user has seen are partitioned into two datasets: a training set (4/5) and a test set (1/5). Before proceeding with multicriteria rating technique, we first observe how single-criteria recommender systems make predictions and recommend movies.

3.2.1 Procedure for Single Criteria Recommendations

In single criteria recommendation approach, cosine-based similarity is used to measure similarity of active user with other users.

We can calculate cosine-based similarity between user *u and u'* as follows:

$$sim_{i,j} = \frac{\left(\sum_{k \in I(i,j)} r_{i,k} r_{j,k}\right)}{\left(\sqrt{\sum_{k \in I(i,j)} r_{i.k}^{2}} \sqrt{\sum_{k \in I(i,j)} r_{j,k}^{2}}\right)} \tag{3.6}$$

$I(i.j)$ represent the set of movies rated by both users i and j.

$r_{i,k}$ represent rating by user i for item k.

$r_{j,k}$ represents rating by user j for item k.

Using formula (3.6) , we get a similarity measure varying from 0 to 1 that represents how similar the user j is to the active user i. Here value 0 represents completely dissimilar users and value 1 signifies exactly similar users.

Main steps:

- Obtain predicted rating ($pr_{i,k}$) for each movie in the test set for active user based on the following prediction formula :

$$pr_{i.k} = \bar{r}_i + \frac{\sum_{j \in N(i)} sim_{i,j}.(r_{j,k} - \bar{r}_j)}{\sum_{j \in N(i)} |sim_{i,j}|} \tag{3.7}$$

$\bar{r}_i$ and $\bar{r}_j$ represents average rating for user i *and* j and j belongs to the neighborhood set of i.

- Now, for each movie in the test set:
  - Compute the error i.e. $(pr_{i,k} - r_{i.k})$ for each movie.
  - Compute Mean Absolute Error (MAE) for active user u_i based on the formula:

$$MAE(u_i) = \frac{1}{|S_i|} \sum_{k=1}^{|S_i|} |pr_{i,k} - r_{i,k}| \tag{3.8}$$

where S_i is the cardinality of the test ratings set of user u_i. The total MAE over all the active users, N_T can be calculated as

$$MAE = \frac{1}{N_T} \sum_{i=1}^{N_T} MAE(u_i) \qquad (3.9)$$

MAE value is used for comparing single and multicriteria rating systems. Lower MAE corresponds to more accurate predictions of RS technique.

3.2.2 Collaborative Filtering with Multicriteria Ratings

In the case of multicriteria ratings also, we divided the data into two sets: training set (4/5) and test set (1/5) as in single criteria recommendations. Neighborhood formation involves training set and rating predictions are made using test set. A four step approach used to generate recommendations based on multicriteria ratings for movie application is explained below :

Step 1: Predict multicriteria ratings. First we consider the multicriteria rating problem of movie recommendation with four criteria as 4 single - rating recommendation problems and estimate unknown ratings for each individual criterion in the same way as in single criteria recommendations done in section 3.2.1. This step provides estimated ratings for all the four criteria (r_1', r_2', r_3', r_4') for each movie in the test set of active user [Adomavicius et al., 2007].

Step 2: Feature weights. W_S, W_A, W_D, W_V can be obtained using any of the techniques given below depending on the availability of data:

- Eigen Vector method (details in section 3.1.1)
- Genetic Algorithms (details in section 3.1.2)

Step 3: Predict overall ratings. We compute overall rating r_0' for each movie in the test set directly by using formula (3.1) i.e.

$r_0' = W_S r_1' + W_A r_2' + W_D r_3' + W_V r_4'$

Step 4: Compute MAE. We first compute error $(pr_{i,k} - r_{i,k})$ for each movie and averaging the error for all the movies in the test set gives Mean Absolute Error (MAE) i.e.

$$MAE\ (u_i) = \frac{1}{|S_i|}\sum_{k=1}^{|S_i|}\left|pr_{i,k} - r_{i,k}\right|$$

Total MAE over all the active users N_T is calculated using formula (3.9) i.e.

$$MAE = \frac{1}{N_T}\sum_{i=1}^{N_T} MAE(u_i)$$

Chapter 4

EXPERIMENTS AND RESULTS

We have conducted two experiments for computing weights (using Eigen vector method and Genetic Algorithms) and for Collaborative Filtering using multicriteria ratings. In order to conduct our experiments, we are using data set obtained through survey , for 100 users and 25 movies with four criteria i.e. story , acting , direction and visuals .

4.1 Experiment to compute criteria weights

In this experiment we compute weights for four movie criteria through two techniques :

a) **Eigen Vector method** : Eigen vector method is used for computing feature weights as described in section 3.1.1. In this experiment we implement Eigen vector method over entire users' database.

The system first compute feature weights for four criteria of movies for all the users. Thereafter it picks the movies, from the data set of each user , one by one and predicts overall rating r_0 for them using formula

$$r_0 = W_S r_1' + W_A r_2' + W_D r_3' + W_V r_4'$$

Mean Absolute Error is computed for each active user using formula

$$MAE\,(u_i) = \frac{1}{25}\sum_{k=1}^{25}\left|pr_{i,k} - r_{i,k}\right|$$

b) **Genetic Algorithm Approach :** An elitist genetic algorithm is used in this experiment for evolving features' weights using movie dataset. A simple unsigned binary encoding is used in this implementation, using 3-bits for each of the 4 feature. The GA begins with random genotypes and evolve weights through method described in chapter 3, section 3.1.2.

A supervised learning is used to learn weights. The GA learns weights using guidance of the actual ratings in the data set of each user. The evolved set of weights is employed to find the predicted ratings for each movie in the data set. The average of the difference between the actual and predicted ratings of all the movies in the data set is used as a fitness score for the set of weights for each user .

4.1.1 Analysis of the results

Both of these feature weighting techniques work on entirely different type of data. Any of these methods could be used to compute weights depending on the availability of data but the data required for Eigen vector technique is not readily available. The computational complexity is less in case of Eigen vector method as compared to GA approach but GA performs better in terms of accuracy. Mean absolute error produced by Eigen vector method are compared against fitness score obtained through GA to check the accuracy of weights. Figure 4.1 and 4.2 show MAE obtained from Eigen Vector method and GA for two different sets of fifty users.

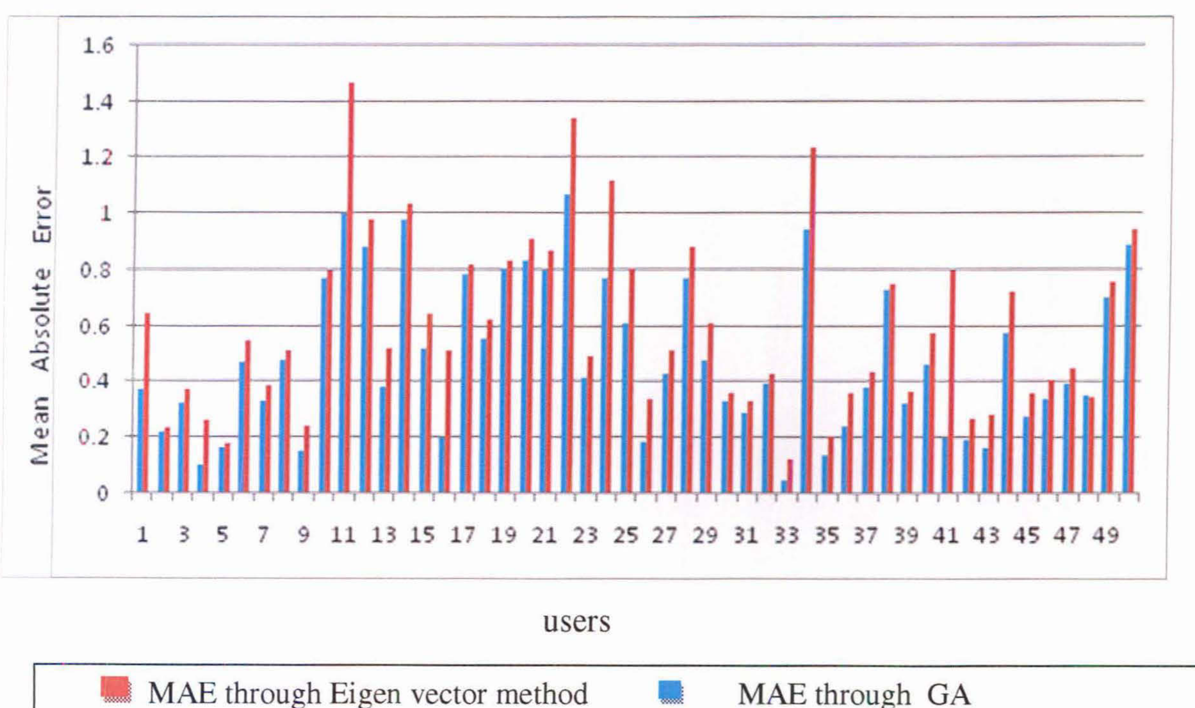


Figure 4.1 Mean Absolute Error for the first set of 50 users

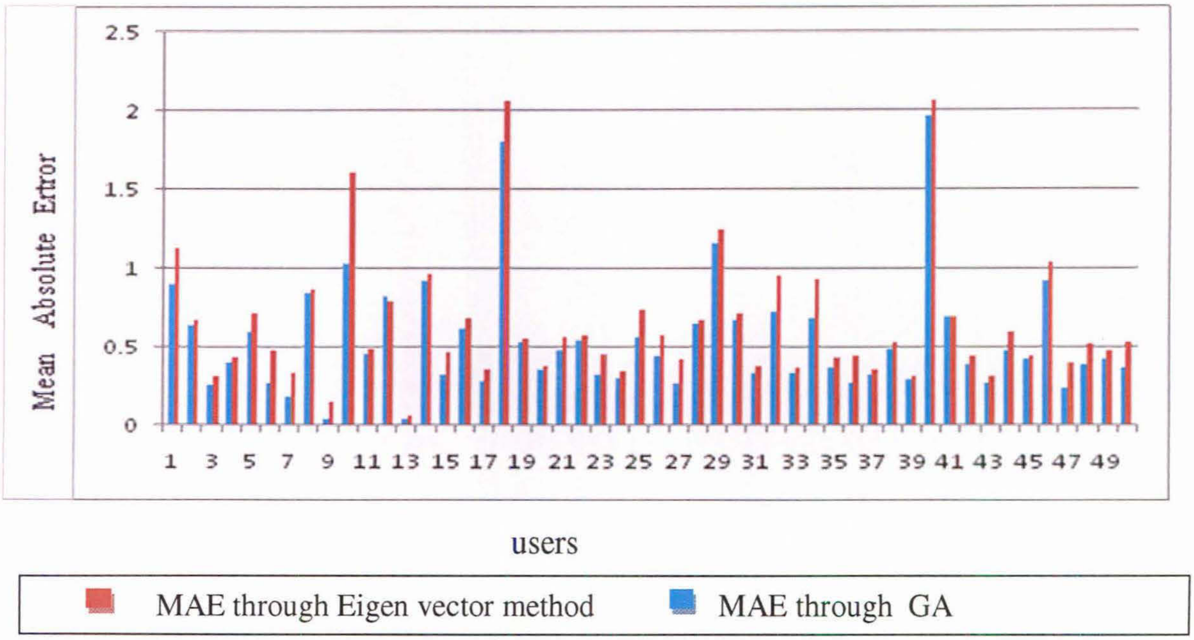


Figure 4.2 Mean Absolute Error for the second set of 50 users

Clearly accuracy of GA approach is higher than that of Eigen vector method.

4.2 Experiment to compare Single criteria and Multi-criteria Recommendations

In this experiment we compared single criteria recommendations with multi-criteria recommendations. Based on the movie dataset we have used 15 movies to build a user model and 10 for testing. We have considered 90 users out of 100 who have rated at least 13 movies. This dataset is used as a basis to generate nine random splits into training and active users. For each random split, 10 users were chosen randomly as active users and the remaining 80 users as training users. Such a random separation was intended for the execution of nine- fold cross-validation, where all experiments are repeated nine times, once with each split. These splits are referred to as split-1, split-2,.........., split-9. The set of training users (80 users) is used to find a set of neighbors for the active user while the set of active users (10 users) is used to test the performance of the system. During the testing phase, each active user's ratings are divided randomly into two disjoint sets, training ratings (60%) and test ratings (40%) . the training ratings are used to model the user and to supervise the learning process of single and multicriteria RS , whereas the test ratings are treated as unseen ratings that the system would attempt to predict.

Mean Absolute Error is computed using formula (3.8) and (3.9) to evaluate the effectiveness of single criteria recommender system and multicriteria recommender system. The MAE measures the deviation of predictions generated by the RS from the actual ratings specified by the user.

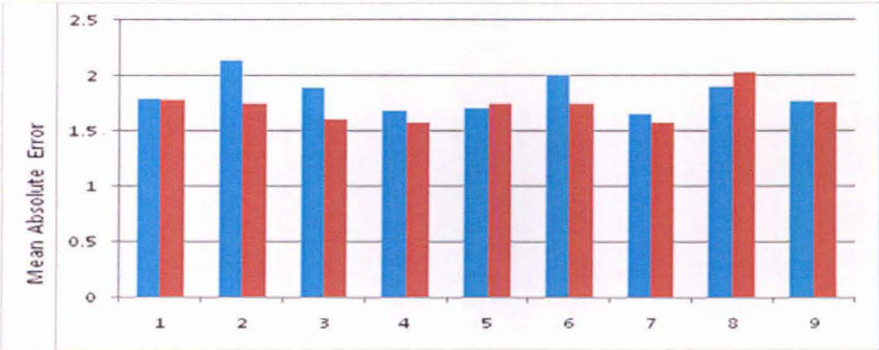
Lower MAE corresponds to more accurate predictions of the given RS. In this experiment, all the nine splits of data are used to show the effectiveness of the proposed collaborative filtering with multicriteria ratings.

4.2.1 Analysis of the Results

The experimental results as given in Table 4.1 and Figure 4.3 show that for 7 out of 9 splits, the multicriteria rating system performs better than the single criteria rating system. Only for split 5 and 8, MAE for single criteria rating is slightly less than that of multicriteria rating system.

Split	MAE for Single Criteria Ratings	MAE for Multi- Criteria Ratings
1	1.79	1.78
2	2.14	1.75
3	1.89	1.61
4	1.68	1.58
5	1.71	1.75
6	2.01	1.75
7	1.65	1.58
8	1.9	2.03
9	1.77	1.76

Table 4.1 : Single Criteria versus Multicriteria Rating Systems



Splits

Figure 4.3 : Single Criteria versus Multicriteria Systems

Chapter 5

CONCLUSION

This dissertation presents a collaborative filtering approach with multicriteria ratings and a comparison between two weight computation techniques for various criteria. In this technique, multicriteria ratings use collaborative filtering approach to predict ratings. Genetic algorithms and Analytical Hierarchical Process are used to compute criteria weights.

Experimental results show that there is considerable increase in the accuracy of prediction by incorporating multicriteria ratings in collaborative filtering. Also Genetic Algorithms gives more accurate weights for various criteria than Analytical Hierarchical Process.

Future Work

Further work is required to test the proposed approach on large datasets and other domains. In the present work we have assumed the aggregation function for the overall rating as a linear combination of the multicriteria ratings and therefore as a future work other aggregation functions need to be tried. One of the future research and directions would be to incorporate multicriteria ratings into Hybrid RS based on compact user model [Al-Shamri et al., 2007] and hybrid user model [Al-Shamri et al., 2008].

REFERENCES

Adomavicius, G. and Kwon, Y. (2007), " New Recommendation Techniques for multicriteria Rating Systems," *IEEE Intelligent Systems, IEEE Computer Society*, pp. 44-55.

Adomavicius, G. and Tuzhilin, A. (2005) , " Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp. 734-749, June 2005.

Adomavicius, G. , Sankaranarayanan, R., Sen , S. and Tuzhilin, A. (2005) , " Incorporating contextual Information in Recommender Systems Using a Multidimensional Approach", *ACM Transactions on Information systems, Vol. 23, No. 1*, pp. 103-145.

Adomavicius, G. and Tuzhilin, A. (2001) , " Multidimensional Recommender Systems: A Data Warehousing Approach" , L. Fiege, G. Muhl, and U. Wilhelm (Eds.) : *WELCOM 2001, LNCS 2232, © Springer-Vrtlag Berlin Heidelberg*, pp. 180-192.

Al-Shamri, M. Y. H. and Bharadwaj, K.K.(2008) , " Fuzzy- Genetic Approach to Recommender Systems Based on a Novel Hybrid User Model ", *Expert Systems and Applications 35-(2008) ELSEVIER* , pp. 1386-1399.

Al-Shamri, M. Y. H. and Bharadwaj, K.K. (2007) , " A Compact User Model for Hybrid Movie Recommender System " , *In Proc. of the 7th IEEE International Conference on Computational Intelligence and Multimedia Applications (ICCIMA ' 07) , Tamil Nadu, India , IEEE Computer Society*, pp. 519-524.

Balabanovic, M. and Shoham, Y.(1997)," Fab: Content-Based, Collaborative Recommendation", *Communications of the. ACM , Vol. 40, No. 3*, pp. 66-72.

Basu, C., Hirish, H. and Cohen, W. (1998), " Recommendation as Classification: Using Social and Content – Based Information in Recommenation " , *In Proc. of the 15th National Conference on Artificial Intelligence, Madison, WI*, pp. 714-720.

Baudisch , P.(2001), " Dynamic Information Filtering" , Ph.D. thesis, Technische Universität Darmstadt, Darmstadt, Germany.

Breese, J. Heckerman, D., and Kadie, C. (1998) , " Empirical Analysis of Predictive Algorithms For Collaborative Filtering," *Proc. 14th Conf. Uncertainty in Artificial Intelligence, Morgan Kaufmann*, pp. 43-52.

Burke, R. (2002) , " Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User- Adapted Interaction 12, Kluwer Academic Publishers,* pp. 331-370.

Burke, R. (2000), " Knowledge-Based Recommender Systems" , *In: a. Kent (ed.):En cyclopedia of Library and Informatin systems, Vol. 69, supplement 32.*

Burke, R. , Hammond, K. and Young, B. (1997) ," The FindMe Approach to Assisted Browsing", *IEEE Expert , Vol. 12(4)*, pp. 32-40.

Claypool, M. , Gokhale , A. , Miranda, T. , Murnikov, P. , Netes, D. and Sartin, M. (1999), " Combining Content - Based and Collaborative Filters in an Online Newspaper " , *SIGIR '99 Workshop on Recommender Systems : Algorithms and Evaluation.* Berkeley, CA.

Drachsler, H. , Hummel, H., and Koper, R. (2007) , " Recommendations for Learners are different: Applying Memory - based Recommender System Techniques to Lifelong Learning" , *In Proc. of the 1st Workshop on Social Information Retrieval for Technology – Enhanced Learning and Exchange SIRTEL '07 , Crete, Greece.*

Forman, E. and Selly, M. A. , (2001) , " Decision By Objectives," *World Scientific.*

Goldberg, D. E. (1989), "Genetic Algorithms in Search, Optimization and Machine Learning", *Pearson Education*, Singapore.

Haq, A. N. and Kannan , G. (2005) " Fuzzy Analytical Hierarchy Process for Evaluating and Selecting a Vendor in a Supply Chain Model" , *Int J Adv Manuf Technol, Vol. 29, © Springer-Verlag London Limited*, pp. 826-835.

Herlocker, J. L. , Konstan, J. A. , Borchers, Al and Reidl , J. (1999) , " An Algorithm Framework for Performing Collaborative Filtering" , *In Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press*, pp.230- 237.

Holland, J. H. (1975), Adaptation in Natural and Artificial Systems, *University of Michigan Press* ,Ann Arbor, Michigan, USA.

Lang, K. (1995) , " NewsWeeder: Learning to Filter Netnews, " *In Proc.of the 12th International Conference on Access* (Stanford, Calif) .

Lee, H.H. and Teng W.G.,(2007), " Incorporating Multi - Criteria Ratings in Recommendation Systems" , *Information Reuse and Integration , IRI '07. IEEE International Conference,* pp. 273-278.

Li, Q., Wang, C. and Geng ,G. (2008) , " Improving Personalized Services in Mobile Commerce by a Novel Multicriteria Rating Approach" , *WWW 2008, Poster Paper*,Beijing, China.

Lim, M.H. , Rahardja, S. and Gwee, B.H. (1996) , " A GA Paradigm for Learning Fuzzy Rules", *Fuzzy Sets and Systems, Volume 82, Issue 2*, pp. 177-186.

Linden, G, Smith, B. and York, J. (2003) , " Amazon.com Recommendations: Item –to - Item Collaborative Filtering' , *IEEE Internet Computing*, pp. 76-80.

Manouselis, N. and Costopoulou, C. (2006a) , " Experimental Analysis of Design Choices in Multi-Attribute Utility Collaborative Filtering, " *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), Special Issue on Personalization Techniques for Recommender Systems and Intelligent User Preferences* , pp. 311-331.

Manouselis, N., Costopoulou, C. (2006b), "Designing a Web-based Testing Tool for Multi-Criteria Recommender Systems", *Engineering Letters, Special Issue on "Web Engineering", 13(3),* (http://www.engineeringletters.com/issues_v13/issue_3/EL_13_3_22.pdf).

Manouselis N., Costopoulou C., (2007) , "Analysis and Classification of Multi-Criteria Recommender Systems", *World Wide Web: Internet and Web Information Systems, Vol. 10 , Issue 4*, pp. 415- 441.

Melville, P. , Mooney , R. J. and Nagarajan, R. (2005), " Content – Boosted Collaborative Filtering for Improved Recommendations," *Proc. of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002*, Edmonton,Canada, *)*, pp. 187-192.

Min, S. H. and Han, I (2005) , " Optimizing Collaborative Filtering Recommender Systems", *P.S. Szczepaniak et al. (Eds): AWIC 2005, LNAI 3528, © Springer - Verlag Berlin Heidelberg*, pp. 313-319.

Mitchell, M.(1996), "An Introduction to Genetic Algorithms",*MIT Press, Cambridge*, MA,U.S.A.

Mitchell, T. (1997), " Machine Learning" , McGraw-Hill.

Mooney , R.J. and Roy, L. (2000), " Content - based Book Recommending Using Learning for Text Categorization" , *In Proc. of the Fifth ACM Conference on Digital Libraries, ACM Press*, pp. 195-204.

Pazzani , M. J. (1999) , " A Framework for Collaborative, Content-Based and Demographic Filtering", *Artificial Intelligence Review*, 13(5/6), 393-408.

Pazzani, M. , Muramatsu, J. and Bilisus,D. (1996) " Syskill & Webert : Indentifying Interesting Websites ", *In Proc. of the 13th National Conference on Artificial Intelligence (AAAI-96) , Portland OR* , pp. 54-61.

Plantié, M., Montmain, J. and Dray, G. (2005) , " Movies Recommenders Systems: Automation of the Information and Evaluation Phases in a Multi- criteria Decision – Making Process" , *K.V. Andersen, J. Debenham, and R. Wagner (Eds.) : DEXA 2005, LNCS 3588, © Springer -Verlag Berlin Heidelberg*, pp. 633-644.

Resnick, P. , Iacovou, N. , Suchak, M., Bergstrom, P. and Reidl , J. (1994) , " Grouplens: An Open Architecture for Collaborative Filtering of Netnews" , *Proceedings ACM 1994 Conference, Computer Supported Cooperative Work, ACM Press* , pp. 175-186.

Resnick , P. and Varian ,Hal R. (1997) , " Recommender Systems",*Communications of the ACM,* 40(3): pp. 56-58.

Rich, E. (1979) , " User Modelling via Stereotypes " , *Cognitive Science, Vol. 3, No. 4,* pp. 329-354.

Schmitt, C. , Dengler, D. and Bauer , M. (2002) ," THE MAUT MACHINE: An Adaptive Recommender System, " *In Proc. ABIS Worksh. ' Adaptivitat und Benutzermodellierung in Interaktivan Softwaresystemen'*, Hannover, Germany.

Smyth, B. and Cotter, P. (2000) , " A Personalized TV Listings Service for the Digital TV Age", *Knowledge-Based Systems , Vol. 13*, pp. 53-59.

Steiguer, J.E. de , Duberstein, J. and Lopes , J. (2003), " The Analytic Hierarchy Process as a Means for Integrated Watershed Management " , *In Renard, Kenneth G., First Interagency Conference on Research on the Watersheds, Benson, Arizona: U.S. Department of Agriculture, Agricultural Research Service,* pp. 736-740.

Ujjin , S.and Bentley, P. (2004) ," Using Evolution to Learn User Preferences" *,In: Tan, M. Lim, X. Yao, and L. Wang, Ed., Recent Advances in Simulated Evolution and Learning, World Scientific Publishing* , pp. 20-40.

Vozalis , E. and Margaritis , K. G. (2006) , " Analysis of Recommender Systems' Algorithms," *Proceedings of the 2006 ACM symposium on Applied Computing, Dijon, France,* pp. 872-878.

Wei, Y.Z. , Moreau, L. and Jennings, N. R.(2005) , " A Market-Based Approach to Recommender Systems," *ACM Transactions on Information Systems, Vol. 23, No. 3,* pp. 227-266.