

A Hybrid Matchmaking Engine Using Advanced Recommender System

*A project report submitted in partial fulfillment of the requirements for
B.Tech. Project*

B.Tech.

by

**K. Shashank (2014IPG-047)
P. Akhilesh Naik (2014IPG-111)
Jk. Snehith (2014IPG-121)**



**ABV INDIAN INSTITUTE OF INFORMATION
TECHNOLOGY AND MANAGEMENT
GWALIOR-474 010**

2017

CANDIDATES DECLARATION

We hereby certify that the work, which is being presented in the report, entitled **A Hybrid Matchmaking Engine Using Advanced Recommender System**, in partial fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the institution is an authentic record of our own work carried out during the period *May 2017* to *September 2017* under the supervision of **Dr. Pramod Kumar singh**. We also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date:

Signatures of the Candidates

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date:

Signature of the Research Supervisor

ABSTRACT

Use of online matrimony for matchmaking is the new norm in India on contemporary to the already existing dating sites with a different notion all together. One fundamental component to realize the perfect match is through ranking of matches. Using content-based approaches we designed two distinct scoring algorithms. This algorithms solve the cold-start problems by being able to provide recommendations immediately based on new user's profiles and preferences. The later part is collaborative filtering which uses the interactions of the similar active users, including the people they like/dislike and are liked/disliked by, to result in reciprocal recommendations. The objective of the following study is to enable users to get their match in a more apt and efficient way for both user as well as the recommender system. So, we also make use of collaborative filtering as to achieve the following above statement.

Keywords: Recommender System, Collaborative Filtering, Content Based, AHP

ACKNOWLEDGEMENTS

We are highly indebted to **Dr. Pramod Kumar Singh**, and are obliged for giving us the autonomy of functioning and experimenting with ideas. We would like to take this opportunity to express our profound gratitude to them not only for their academic guidance but also for their personal interest in our project and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within us. The nurturing and blossoming of the present work is mainly due to their valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. Our mentor always answered myriad of our doubts with smiling graciousness and prodigious patience, never letting us feel that we are novices by always lending an ear to our views, appreciating and improving them and by giving us a free hand in our project. It's only because of their overwhelming interest and helpful attitude, the present work has attained the stage it has.

Finally, we are grateful to our Institution and colleagues whose constant encouragement served to renew our spirit, refocus our attention and energy and helped us in carrying out this work.

(K. Shashank)

(P. Akhilesh Naik)

(Jk. Snehith)

TABLE OF CONTENTS

ABSTRACT	ii
LIST OF TABLES	v
LIST OF FIGURES	vi
1 INTRODUCTION AND LITERATURE SURVEY	1
1.1 INTRODUCTION	1
1.1.1 RECOMMENDER SYSTEMS FOR MATRIMONIAL SITES .	1
1.2 LITERATURE REVIEW	2
1.2.1 MATRIMONY MATCHING RECOMMENDER SYSTEM . .	2
2 DESIGN DETAILS AND IMPLEMENTATION	4
2.1 DESIGN DETAILS	4
2.1.1 OVERVIEW OF MATRIMONIAL SITE	4
2.1.2 DOMAIN OVERVIEW OF PROPOSED RECOMMENDER SYSTEM	5
2.2 IMPLEMENTATION	6
2.2.1 ANALYTIC HIERARCHY PROCESS (AHP)	7
2.2.1.1 HOW THE AHP WORKS	8
2.2.1.2 WHAT IS PAIR-WISE COMPARISON ?	8
2.2.1.3 MAKING COMPARISON MATRIX/RECIPROCAL MATRIX	8
2.2.1.4 PRIORITY VECTORS AND EIGEN VECTORS . .	9
2.2.1.5 HOW TO COMPUTE AHP IN FULL HIERARCHY?	9
2.2.1.6 HOW IS AHP USED IN ALGORITHM ?	9
2.2.2 SCORING ALGORITHMS	10
2.2.2.1 ALGORITHM-1	10
2.2.2.2 ALGORITHM-2	14
2.2.3 COLLABORATIVE FILTERING BASED ALGORITHM . . .	17

3	RESULTS AND DISCUSSIONS	21
3.1	RESULTS	21
3.1.1	RESULTS FOR SCORING ALGORITHM - 1	21
3.1.2	RESULTS FOR SCORING ALGORITHM	22
4	CONCLUSION AND FUTURE WORK	25
4.1	CONCLUSION	25
4.2	FUTURE WORK	26
	REFERENCES	26

LIST OF TABLES

- 2.1 Comparison matrix. 10
- 2.2 Attributes Importance and Possible Values. 11
- 2.3 Possible Ranges and Representation of Age Attribute 12
- 2.4 Assigning of Scores to Candidate Users Based on Responses 19

- 3.1 Profiles of active bride grooms 21
- 3.2 Recommendations for Active Bride 22
- 3.3 Advanced Recommendations for Active Bride 22

LIST OF FIGURES

2.1	Proposed System	6
2.2	Content-based approach for matrimony	7
2.3	Collaborative filtering based approach	8
2.4	AHP Hierarchy	9
2.5	Mapping a Bridegroom Preferences with a Bride's Profile	15
2.6	Recommender Process	17
3.1	User-Bridegroom Preferential List	23
3.2	User-Bride Profile	24

ABBREVIATIONS

RS	Recommender System
MS	Match-making system
AHP	Analytic Hierarchy Process

CHAPTER 1

INTRODUCTION AND LITERATURE SURVEY

This chapter includes the details of recommender systems, concepts of collaborative filtering, purpose of recommender systems in matrimonial sites, how matrimonial sites are different from other domain having various match-making systems, platform used to implement the project and literature review related to work done in this field.

1.1 INTRODUCTION

With the rise in amount of data the choice of selective and required information has always been one of the key problems in today's era.

In such situations recommender systems have paved the way so as to remove the individual burden to make evaluative decision's and reduced overload of information. Almost always the traditional recommender system was used to provide items, information and services in the field of e-commerce.

In many past years Recommendation system that were built utilizing non-personalized, demographic content-based, collaborative filtering and knowledge based RC. Further the recommender system have evolved in areas such as social match-making system where a user is meant to meet other user in order to satisfy a particular need. Many match-making systems that exist today include dating services, resume/job bulletin board, consumer to consumer market places.

1.1.1 RECOMMENDER SYSTEMS FOR MATRIMONIAL SITES

The role of recommender systems have always played a important part of online dating site as social matchmaking RS which dated back to 1965. During the year 1965 a team of Harvard undergrads created Operation Match, the world's first computer dating service. For 3 Dollar, users could answer questionnaires and receive a list of potential

matches, a process that is still used by many dating sites. Online dating sites is a system that enables a user to find and introduce themselves to opposite gender user usually with the goal of developing personal, romantic relationship with a wide variety of unmodulated match-making services. Online dating is the new norm for introductions, replacing the role of traditional personals and in many cases, merging with the functions of social media.

Matrimonial websites are a variation of the standard dating websites popular in India and among Indians settled overseas as an alternative to the traditional marriage broker system that prevailed before online matrimonial sites emerged. Few to name are Bharatmatrimonial.com, shaddi.com and few other Pioneers in the field of social match-making. Despite the popularity of such systems, relatively little research is done on match-making RS that are used in matrimonial sites

1.2 LITERATURE REVIEW

The field of recommender systems has garnered a lot of attention in both the industry and academia, with a considerable volume of literature(1). It is a prevalent way of solving the information rich problems. They have been popularized by widely popular commercial services such as Netflix(2) or Amazon(3). Other popular services exploiting the benefits of RS are match.com (4), Ringo(5), Bellcore (5), Pandora, Tripadvisor, PHelp (4), i-Help (4), Flipkart, Twitter, Facebook etc. Deshpande *et al.* (6) performed an analysis of various term-based top-n recommendation algorithms on several popular datasets including movies, credit card transactions, and others using different metrics.

1.2.1 MATRIMONY MATCHING RECOMMENDER SYSTEM

Deployment of RS in matrimony is an upcoming area of research. Even though matrimony is considered as a potential application for RS there is a surprising lack of literature published on this problem. Considerable amount of work is done in match-making in dating services. A combination of content-based RS and reciprocal algorithm has been implemented in Recon (4) system. User profiles and interactions with other users are used by Recon system as their parameters. This system tried to extract user's onsite preferences which are inferred from his/her communication with users of opposite gender and all the similar profiles are matched with them. Two recommendation engines based on collaborative filtering approaches for an online dating service were developed by Brozosky and Petricek(7) implemented two RS for online dating based on collaborative filtering and tested their algorithm on a real dating service data. The results from their analysis have shown that collaborative filtering based recommender systems outperform popularity based recommender systems. L. Li and T. Li

have implemented MEET (8) system for online dating using bipartite graph. A hybrid content-collaborative approach (9) for online dating was put forward by J. Akehurst *et al.*

K. Joshi and S. Kumar (10) were able to use analytical hierarchy process in matrimony to calculate weights for different attributes. A framework (11) for implementation of collaborative filtering in matrimony was provided by Sampath.M.K *et al.* S. Khavate and M. A. Potey (12) came up with partial fuzzy based scoring system for Indian matrimony.

CHAPTER 2

DESIGN DETAILS AND IMPLEMENTATION

In this chapter we would like to give an overview of matrimonial site, to analyze and understand the concept of the proposed recommender system and the implementation of algorithm using user rank method and the use of collaborative filtering is explained in the latter part of this chapter.

2.1 DESIGN DETAILS

2.1.1 OVERVIEW OF MATRIMONIAL SITE

Unlike the online dating sites where the users preferences are almost accounted for and a small profile has to be filled to choose your match. The matrimonial site is different because of the intent, purpose and the requirement that the user wants. Here, users range from self to serious parents and grandparents, search each and every recommendation with at-most care in hope to find the perfect match for the bride or groom in every recommendations provided to them give.

First, the user of matrimonial sites have to create a profile for himself or the person for whom he/she would want to find a match. This is done by registering into it and giving detail on one's own free will. They must also provide preference details about the partner. User profile will be generated once the user submits the registration form. The user now becomes active user and his preferences about their partner are then taken and matches are identified using collaborative filtering. Now, the user interests are identified by using personalized algorithms and profiles when matched user interests are identified. When the profiles that satisfy both users preferences and interest are now considered for recommendation from this we obtain top n-profiles based on weightage of profiles to each active user. One of the main problem of having only collaborative filtering or reciprocal recommender system is it does not specify the degree of accep-

tance of one match over other and is inefficient in terms of giving recommendation to all the active user.

User's profile is a union of both female and male user's in a matrimonial sites now these profiles are further divided into account information, contact information, personal profile and preferred partner's profile. Account information comprises of login name, password, user's name, date of birth and security question and answer. Contact information normally consist of address, city, state, country, mobile number and email-id of the user. Personal profile consists of attributes like age, height, religion, caste, education, occupation, income, complexion, body type, diet, smoke and drink. The preferred partner profile consist of user's excepted range of age, range of height, range of income, religion, caste, education, occupation, complexion, body-type, diet, smoke and drink. Some times user may not put any preferences to a particular attribute and this means the whole range is said to be accepted such as diet is not preferred to be a specific type then the user's partner can be from an origin having any kind of diet. The other attributes are smoke and drink which are preferred attributes for both female and male.

2.1.2 DOMAIN OVERVIEW OF PROPOSED RECOMMENDER SYSTEM

- 1) Creating *user profile* - New user have to login to the website and give details such as their personal information, contact information and Partner preferences to become an active user.
- 2) The attributes are mainly of 12 types which constitutes as partner's preference list, these 12 attributes are further generalized into 2 groups containing 6 attributes each of HIGH importance and the other 6 attributes of LOW importance.
- 3) Then a preference attribute vector is formed to shortlist the matches in N-recommendations.
- 4) Browsing other active users for interesting matches.
- 5) The users interests on match which is recommended is calculated by the degree of similarity between opposite gender HIGH importance attributes i.e. six attributes which come into the category of HIGH this other LOW category attributes are calculated using Weights of each attribute that the user has given importance to and which in-turn helps in bringing more related recommendations.
- 6) The weights for attributes are given using the AHP.

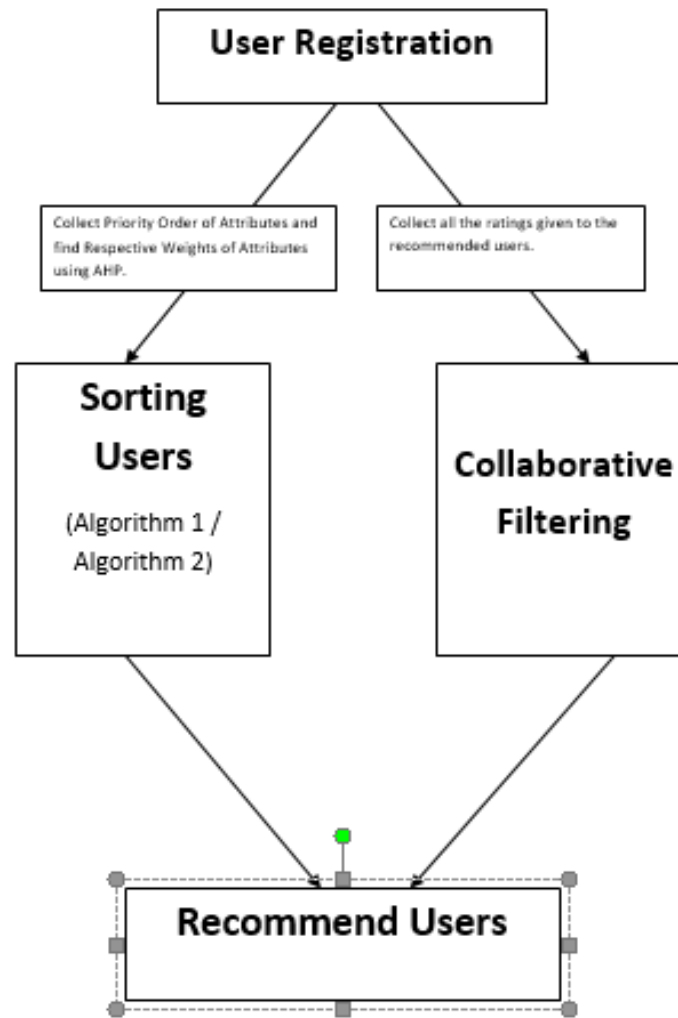


Figure 2.1: Proposed System

7) Now Using *collaborative filtering* - the similarity must be found between the active users and new active users so as to make it efficient by suggesting the same N-recommendation based on degree of similarity in preferences that were provide by both users to find their respective matches.

2.2 IMPLEMENTATION

Content-based and collaborative based filtering are the most common approaches used in recommendation engines. Content-based methods develop user preferences profile based on his/her initial preferences given during signup (e.g. Vegetarian matches in matrimony example) and/or the attributes of the profiles rated positively and/or negatively by the users, the model is based on this data and this model is used to show new profiles that have similar attributes. The techniques based on content-based techniques

usually do not take into account the data from other users, they only take the data of the current user into consideration. However well these approaches may address the cold-start problem, the other kind of approaches (collaborative filtering) are more versatile and efficient. These approaches exploit data from other users who gave similar ratings on the same items and recommend items that have been rated positively by other users and not yet by the target user. Both approaches share their own set pros and cons. In our field of online matrimony, the items to be recommended are human beings, who have their own set of preferences, profile and activity in the site. This poses different challenges in giving recommendations as it is considered important to recommend useful recommendations to newly joined users, so that they don't feel bad at the very beginning, hence it becomes critical to solve the cold-start problem. So we decided to come up with two types of algorithms serving different purposes with a common goal to increase user satisfaction by recommending good matches. The two scoring algorithms discussed in the subsequent section falls in the category of content-based filtering. They only use the preferences of the user while giving recommendations to that user. It is followed by an algorithm based on collaborative filtering which uses the ratings given by other users.

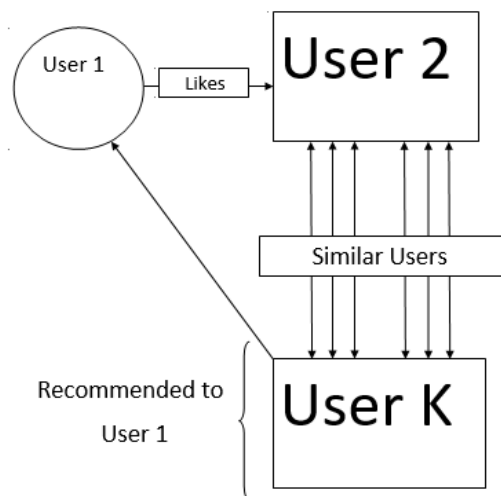


Figure 2.2: Content-based approach for matrimony

2.2.1 ANALYTIC HIERARCHY PROCESS (AHP)

The Analytic Hierarchy Process (AHP) is an effective tool for dealing with complex Multi-Criteria decision making framework and assists the decision maker to derive ratio scales from paired comparisons or by setting priorities and synthesize the results. The AHP can intake both subjective measurement such as satisfaction feelings, preference and objective/Actual measurements such as price, weight etc of a decision. In

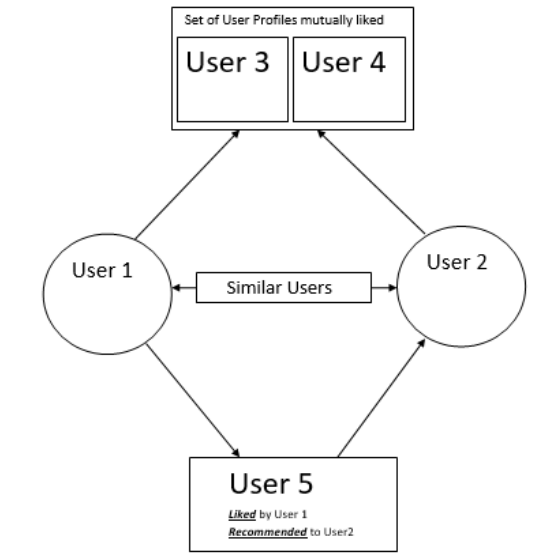


Figure 2.3: Collaborative filtering based approach

addition, the AHP includes a useful technique for checking the conformity or regularity of the decision maker's evaluations, thus reducing the prejudice in the decision making process.

2.2.1.1 HOW THE AHP WORKS

The AHP takes a set of evaluation attribute into account. The AHP generates a weight for each evaluation attribute according to the decision maker's pairwise comparisons of the attribute. The higher the weight, the more important the corresponding attribute.

2.2.1.2 WHAT IS PAIR-WISE COMPARISON ?

Pairwise comparison generally is any task of comparing attribute in pairs to judge which of each attribute is favorable or has a greater amount of some quantitative property or whether or not the two attribute are identical. Two attributes are computed at a time in terms of their comparative importance. Index values from 1 to 9 are used.

2.2.1.3 MAKING COMPARISON MATRIX/RECIPROCAL MATRIX

Here the Pairwise comparisons are analyzed. If attribute **A** is exactly as significant as attribute **B**, this pair receives an index of 1. If **A** is prime than **B**, the index is > 1 . A Spectrum of values from 1 to 9 are possible. For a "less important" relationship, the fractions from $1/9$ to $1/2$ are available. The values are recorded row by row into a cross-matrix of size (number of attributes considered \times number of attributes considered). All the diagonal elements contains the value 1. If a Attribute A_i to A_j was rated with corresponding importance of index n , A_j to A_i has to be rated with index $1/n$.

2.2.1.4 PRIORITY VECTORS AND EIGEN VECTORS

Having a comparison matrix i.e. priority vector, which is the normalized Eigen vector of the matrix is computed. The weights of the individual criteria are calculated. First, a normalized comparison matrix is created: each value in the matrix is divided by the sum of its column. The normalized principal Eigen vector can be obtained by averaging across the rows. These weights are already normalized and their sum is 1. The normalized principal Eigen vector is also called priority vector (sum of all elements in priority vector is 1).

2.2.1.5 HOW TO COMPUTE AHP IN FULL HIERARCHY?

Level 0 is the objective of the analysis. Level 1 is multi attribute that consist of various other attributes or parameters. Diversifying into many levels of sub criteria and much lower levels of attribute or alternatives is also possible. The last level of the hierarchy is the alternative options. The lines or links indicates relationship between Goals, Factors (Attribute level), Alternatives (Sub-Attribute level). In Level 1 a priority or comparison matrix corresponds to pair-wise comparisons between 4 Attributes with respect to goal. Thus a priority matrix of size 4 is prompted. Upon examining the Alternative / Sub-Criteria level i.e. Alternatives of Attribute 1 / Criterion 1, a pair-wise comparisons between 3 Alternatives is accounted. Consequently a priority matrix of size 3 for every attribute in Level 1 is created. Finally, the final weightage of a Alternative is concluded by multiplying the weightage score of its corresponding Attribute with the weightage of the Alternative in its level.

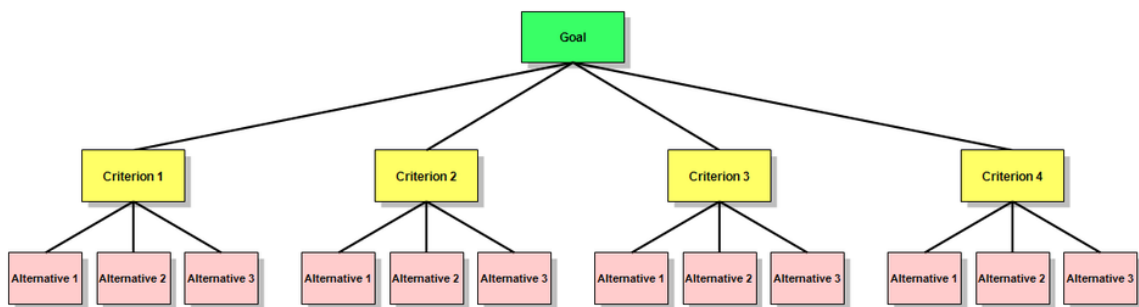


Figure 2.4: AHP Hierarchy

2.2.1.6 HOW IS AHP USED IN ALGORITHM ?

Firstly, our algorithm takes the priority order of attributes and makes a comparison matrix M of size $n \times n$. The value of element M_{ij} is (distance between Attribute A_i and Attribute A_j in the priority order + 1) if and only if Attribute A_i is prior to Attribute A_j .

If Attribute A_j is prior to Attribute A_i then value of element M_{ij} is $1/(\text{distance between Attribute } A_j \text{ and Attribute } A_i \text{ in the priority order} + 1)$. After obtaining M_{ij} , M_{ji} is calculated by inversing M_{ij} .

For Example,

Algorithm takes the the priority order of the user as a, b, c, d which technically mean $a > b > c > d$ then the corresponding comparison matrix of size 4×4 is as follows : corresponding weights:

Table 2.1: Comparison matrix.

	a	b	c	d
a	1	2	3	4
b	1/2	1	2	3
c	1/3	1/2	1	2
d	1/4	1/3	1/2	1

$Weight_a = 0.46$, $weight_b = 0.27$, $weight_c = 0.15$, $weight_d = 0.09$

2.2.2 SCORING ALGORITHMS

2.2.2.1 ALGORITHM-1

This algorithm consists of following steps:

Classification of Attributes:

The main function of a match-making recommender system is to recommend profiles considering what user primarily wants, and thus increase user's overall satisfaction. To achieve this attributes are divided into two groups. This classification is based on the importance of attributes. The first group consist of attributes of high importance like religion, caste, occupation, diet, marital status, smoke and drinking habits. Other group consist of less important attributes like age, height, education, annual income, body type and complexion. Our algorithm makes sure that all the attributes belonging to the first group are perfectly matched while giving recommendations. And it then uses the less important attributes in scoring profiles. TABLE 2.2 shows each of these attributes along with its importance and some possible values that attribute would take. New possible values for any attribute can be added into implemented system.

Table 2.2: Attributes Importance and Possible Values.

Sl. No.	Attribute	Importance	Possible values
1.	Age	Low	20 to 35
2.	Height	Low	4.8 to 6.3
3.	Religion	High	Hindu,Jain,Christian, Parsi,Muslim,Sikh, Buddhist,Jewish
4.	Caste	High	Arya,Aggarwal, Rajput,Sindhi, Patel,Catholic, Parsi,Buddhist, Jewish,Pama,Teli, Maratha,Brahmin, Sunni,Shia, Digam-ber, Vaishnav, Protestant
5.	Education	Low	Doctors(PhD,Mphil), Mas-ters(ME,MTch,MSc, MS,MD,MCm,MA), Bachelors(BE, BTech,BCom,BA, BSc,BHMS,BAMS, MBBS,BDS,LLB), Diploma(Diploma, DMLT),HSC,Below HSC(SSC,9th)
6.	Occupation	High	Assistant, Govern-ment, Businessper-son, Consultant, Designer,Doctor, Lawyer,Principal, Teacher,Professor, Researcher, Lec-turer, Lab technician
7.	Annual In-come	Low	100,000Rs to 1,500,000 Rs
8.	Body Type	Low	Slim, Athletic, Aver-age, Heavy
9.	Complexion	Low	Veryfair, Fair, Wheatish, Dark
10.	Diet	High	Non-veg, Veg
11.	Smoke	High	Yes, No
12.	Drink	High	Yes, No

The possible values of age, height and income attributes are distributed into ranges. In our system we considered eight possible ranges for each of these three attributes. One possible range allocation for age attribute is shown in TABLE 2.3.

Table 2.3: Possible Ranges and Representation of Age Attribute

Age	Representation
20-22	1
22-24	2
24-26	3
26-28	4
28-30	5
30-32	6
32-34	7
34-	8

For instance, a new female user has registered into the matrimonial site and she has specified partner preference attributes value as follows: Age - 26 to 28, Height - 5.9 to 6, Religion - Hindu, Caste - Doesn't Matter, Education - BTech, Occupation - Engineer, Annual Income - 800,000 Rs. to 1,200,000 Rs., Body Type - Athletic, Complexion - Fair, Diet - Non-Veg, Smoke - No, Drink - No. Then her preference attribute vector would be set as [5, 6, 1, 0, 4, 10, 4, 2, 3, 2, 1, 1]. Doesn't Matter is taken as 0.

Shortlisting Users for N-Recommendations.

This step produces an unordered list of recommendations for a user. The active user's preference attribute vector is compared against each of the profiles in opposite gender and a list of recommendations is generated where all the attributes of high importance matches perfectly. By doing this we avoid computing scores for all the users in the database, and also computational load on the system can be minimized as only few users will be considered for scoring discussed in the next step. This ensures that the n-recommendations given by our algorithm satisfies or suits the requirements of Indian marriage culture.

Calculation of Similarity Index Between Two Users:

The unordered list of N-recommendations generated in the previous step doesn't fulfill the complete purpose of a good recommender system (RS). In order to meet the personalized preferences of any user, a good RS must be able to display recommendations in some order unique for that user. This is discussed in this step.

As we have already used six high important attributes in the previous step of shortlisting recommendations, we will use the remaining six low important attributes in this step to achieve our purpose. Each of these six attributes contribute to the overall

matching-score for a user-user comparison.

For each of these six attributes, a similarity index is calculated between active user's preference value and other user's profile value for that attribute. This similarity index is multiplied by weight corresponding to that attribute.

For example, if we consider the attribute complexion, the four possible values for that attribute are *Very Fair*, *Fair*, *Wheatish* and *Dark*. Now if active user *A*'s complexion preference is *Wheatish* and shortlisted users *B*, *C*, *D*, *E* preferences for that attribute are *Dark*, *Wheatish*, *Fair* and *Very Fair* respectively. Similarity index for each comparison is calculated by the formula:

$$\text{Similarity Index} = 1 - \text{dist} * \text{factor}$$

Where *dist* is the distance between active user's preference value and other users profile value for that attribute

factor is

$$1/\text{no of possible values for that attribute}$$

Now similarity indices for complexion

$$A - C \text{ is } 1 - 0 = 1$$

$$A - B \text{ and } A - D \text{ is } 1 - 1 * 0.25 = 0.75$$

$$A - E \text{ is } 1 - 2 * 0.25 = 0.5$$

This method doesn't consider user's onsite behavior when calculating similarity-index, it just relies on user's preferences taken from him/her during signup. This can be accomplished by using user-user ratings data. For the same example discussed earlier, if user *A* has previously liked 100 users and among them 20 users falls under *Dark* complexion, 40 under *Wheatish*, 30: *Fair* and 10: *Very Fair*. Now the similarity index is calculated as

$$A - C \text{ is } \frac{40}{40} = 1$$

$$A - B \text{ is } \frac{20}{40} = 0.5$$

$$A - D \text{ is } \frac{30}{40} = 0.75$$

$$A - E \text{ is } \frac{10}{40} = 0.25$$

This process is repeated for other five less important attributes to calculate corresponding similarity index.

Aggregation Method and Ranking Users

In this step, a final score is assigned to each user. For each less important attribute, similarity index is multiplied by attribute weight. Summation of all these multiplications gives the score for that particular user.

$$Score = \sum_{i=1}^6 similarity_i * weight_i$$

The shortlisted list is rearranged in descending order of their scores and ranks are assigned to them. Users are recommended to the active user in the order of these ranks.

2.2.2.2 ALGORITHM-2

Here we intake *User Profile*, *User Preferential List*, *User Priority Order of Attributes and Order of Criteria*. User priority order of attributes and order of criteria are sent to AHP weighing algorithm to estimate the weights of the attributes. User Profile consists of information details of the user. User Preferential list consists of what attribute values are mostly preferred, lesser preferred and least preferred expecting from a opposite gender user.

Here, 10 Attributes are considered on the 3 different criterion :

Education/Occupation

(Educational Qualification, Occupation/Profession)

Wealth

(Earnings, Family Status, Location)

Personality

(Age, Height, Complexion, Body type, Diet)

Example :

Attribute : Diet

Existing Values : Eggitarian, Vegetarian, Non Vegetarian

If a user prefers Vegetarian the most and ready to compromise if the user is Eggitarian rather than Non Vegetarian. Then he fills the details as follows :

Most Preferred: Diet : Vegetarian

Preferred: Diet : Eggitarian

Least Preferred: Diet : Non Vegetarian

Upon assigning similarly for 10 other attributes into 3 categorial distribution of Preferences, we have a complete list of User Preferential list.

Let, W_{complxn} be weight of complexion and W_{diet} be Weight of Diet

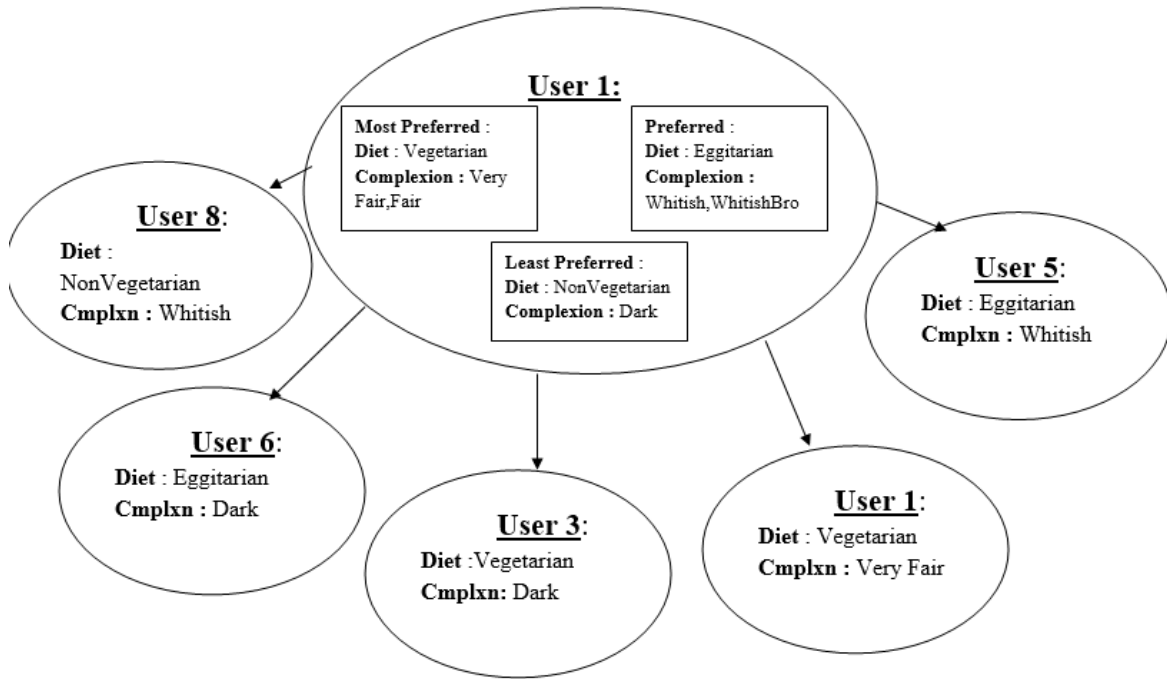


Figure 2.5: Mapping a Bridegroom Preferences with a Bride's Profile

Assume,

$$W_{\text{complxn}} = 0.35, W_{\text{diet}} = 0.01$$

Score generated according to User1 preferential list :

Level 1: (1, 100, 10000)

$$\text{User1} : 10000 * W_{\text{diet}} + 10000 * W_{\text{complxn}} = 3600$$

$$\text{User3} : 10000 * W_{\text{diet}} + 1 * W_{\text{complxn}} = 100.35$$

$$\text{User5} : 100 * W_{\text{diet}} + 100 * W_{\text{complxn}} = 36$$

$$\text{User6} : 1 * W_{\text{complxn}} + 100 * W_{\text{diet}} = 1.35$$

$$\text{User8} : 1 * W_{\text{diet}} + 100 * W_{\text{complxn}} = 35.01$$

Recommendation Order (Top 3) :

$$\text{Scoreuser1} >>> \text{Scoreuser3} > \text{Scoreuser5} == \text{Scoreuser8} > \text{Scoreuser6}$$

Here, algorithm works in such a manner that it tries only to show the users whose most preferred attribute value is matched i.e. by multiplying 10000 to the weightage of the attribute whose value gets matched.

Level 2 : (1, 10, 100)

$$User1 : 100 * W_{diet} + 100 * W_{complx} = 36$$

$$User3 : 100 * W_{diet} + 1 * W_{complx} = 1.35$$

$$User5 : 10 * W_{diet} + 10 * W_{complx} = 3.6$$

$$User6 : 1 * W_{complx} + 10 * W_{diet} = 0.45$$

$$User8 : 1 * W_{diet} + 10 * W_{complx} = 3.51$$

Recommendation Order (Top 3) :

$$Scoreuser1 > Scoreuser5 == Scoreuser8 > Scoreuser3 > Scoreuser6$$

Algorithm tries to show the best matching user profiles from preferred category besides most preferred category.

Level 3 : (1, 4, 16)

$$User1 : 16 * W_{diet} + 16 * W_{complx} = 5.76$$

$$User3 : 16 * W_{diet} + 1 * W_{complx} = 0.51$$

$$User5 : 4 * W_{diet} + 4 * W_{complx} = 1.44$$

$$User6 : 1 * W_{complx} + 4 * W_{diet} = 0.39$$

$$User8 : 1 * W_{diet} + 4 * W_{complx} = 1.41$$

Recommendation Order (Top 3) :

$$Scoreuser1 > Scoreuser5 === Scoreuser8 > Scoreuser3 == Scoreuser6$$

Algorithm tries to show the matching user profiles from preferred category which user may like in addition to most preferred category.

Level 4 : (1, 2, 4)

$$User1 : 4 * W_{diet} + 4 * W_{complx} = 1.44$$

$$User3 : 4 * W_{diet} + 1 * W_{complx} = 0.39$$

$$User5 : 2 * W_{diet} + 2 * W_{complx} = 0.72$$

$$User6 : 1 * W_{complx} + 2 * W_{diet} = 0.37$$

$$User8 : 1 * W_{diet} + 2 * W_{complx} = 0.71$$

Recommendation Order (Top 3) :

$$Scoreuser1 > Scoreuser5 === Scoreuser8 > Scoreuser3 === Scoreuser6$$

If user did not like any of the matches from prior levels then algorithm tries to show the best and better least preferred matches.

2.2.3 COLLABORATIVE FILTERING BASED ALGORITHM

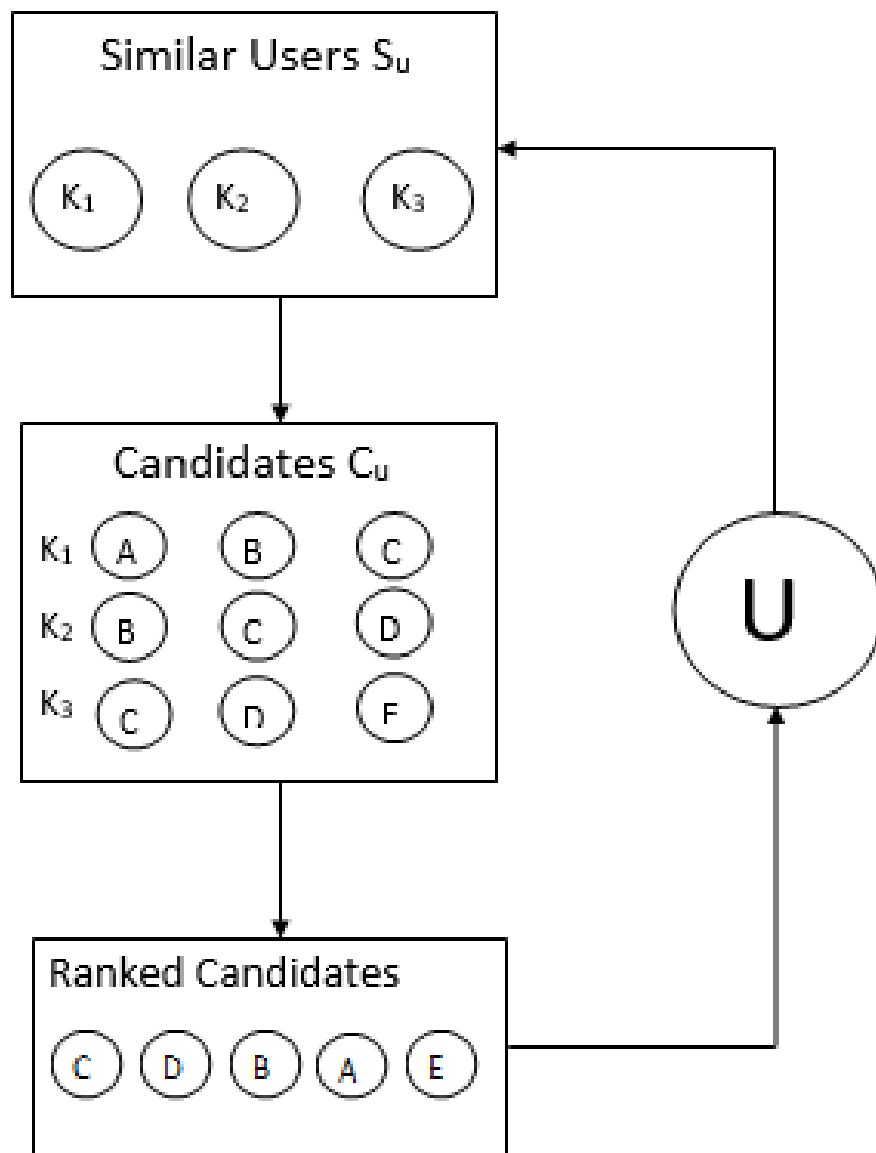


Figure 2.6: Recommender Process

This collaborative approach calculates the similarities between same gender users relying on their preferences and then gives out recommendations from other gender that are expected to be successful in reciprocal order, based on the dislikes and likes

of users in the similar list. By reciprocity we check if both the users being considered like each other or not. For example, if we consider a person U , we list out the users that U is interested in, who also share interest in U . This means U likes someone after seeing that person in his recommendations list and the response of this person was also positive.

The entire process of making recommendation in an ordered fashion comprises of three vital steps, see Figure 2.5. First, we find list the users S_u belonging to same gender group as U and are similar to U by calculating the similarity co-efficient between the preference attributes. Then secondly, we examine the interactions of the users in S_u with users from other gender and generate candidate recommendations C_u . Third, using reciprocity as the scoring parameter we assign ranks to users. These three steps are further discussed below.

Generating Similar Users Based on User Preferences

This is the first step in collaborative filtering approach, finding the similar users. If U is the user under consideration, this step produces a list of K most similar users S_u to U in his/her own gender group. We take user U 's all twelve attribute preferences and find users with similar preferences.

Distance metrics like Manhattan distance, Euclidean distance, cosine similarity, etc., can be employed for this purpose. More the distance between users, less the similarity between them. If the distance between two preferences vectors is 0, then that two persons are said to be identical. For our research, we have used Manhattan distance to calculate the distance.

$$Distance(A,B) = \sum_{i=1}^{12} PrefA_i - PrefB_i$$

Where $PrefA_i$ is i 'th preference of user A

Generating Candidates to be Considered for Recommendation Based on User Interaction

The next step is to generate a list of candidate C_u . This list consists of all candidates to be considered for recommendation in an unordered fashion. Reciprocity is taken into account in deciding the users to be added into this list. For every user in similar users list (S_u), we draw out all the users that he/she shares a reciprocal interest with. This means for a user to join candidate list, he/she has to like any user in similar list and that user in S_u should also like her back. All the users who follow this criteria are included in Candidate list (C_u) and become candidates for recommendation.

For example, in Figure 2.5 if $K1$, $K2$ and $K3$ are the similar users, $[A, B, C]$ is the set of users with reciprocal interest for $K1$ is, $[B, C, D]$ for $K2$: and for $K3$ it is $[C, D, E]$,

then the union, $C [A, B, C, D, E]$ gives the recommendation users list for that user. To aggregate the first two steps, we start with a maximum distance threshold of 1. For a given user U , we calculate his preferences similarity co-efficient with all the users in same gender group and pick the K most similar users into the similar list S_u . Then for each user in S_u , we add all the users he/she was reciprocally interested in to the candidate list. In the next step discussed below, each user in this candidate list is assigned a score based on his/her degree of reciprocity.

Assigning Scores to Candidate Users

The next step is to rank the users in candidate in the candidate list C_u so that we can recommend to the user in that order. This can be achieved by assigning a level of support for all the users in candidate list by considering interactions between the users in similar list and the users in candidate list C_u . The criteria for any user's addition to candidate list is that they have to be reciprocally liked by at least one of the users in similar list. However, some candidates might have received positive interest from more than one S_u user and responded to some positively and to others negatively. Thus, some candidates have more successful interactions with S_u than others. Our ranking approach, uses this multiple reciprocity and calculates score for each user in candidate list and then candidates can be recommended to the user in descending order of their scores.

For each candidate user X in the candidate list, we calculate the number of times X has a positive interaction with a user in similar list S_u . This included the events where he responded positively or received a positive response from a user in S_u , see Table 2.4. Now we calculate the count of negative interaction, number of times X has responded negatively or has received a negative response from S_u . The number of positive minus the number of negative interactions of a user give his/her score in this system. The higher the score for user, the more he is reciprocally liked by users in S_u . The candidates are sorted in descending order based on their score and recommended to the user.

Table 2.4: Assigning of Scores to Candidate Users Based on Responses

Candidate	Positive responses	Negative responses	Score
A	2	7	-5
B	6	5	1
C	12	6	6
D	6	2	4

Thus it can be ensured that users in candidate list who are reciprocally most liked by more number of users in similar users list are always ranked ahead of candidates who are comparatively less liked. At the same time it also accounts for the over represen-

tation of popular candidates as they are more likely to have a higher negative response rate.

CHAPTER 3

RESULTS AND DISCUSSIONS

3.1 RESULTS

In the experimental setup, the proposed system is used to recommend exact or near exact matches who best satisfy the active user's requirements. Evaluation metrics like precision, recall and f-score can be used to evaluate the results of the recommender system.

3.1.1 RESULTS FOR SCORING ALGORITHM - 1

In this algorithm, two levels of calculation are done. Firstly, the entire database is searched to shortlist users who would satisfy the active user's high important attributes. Then his/her less important attributes are used along with his weights for those attributes to assign a score to each shortlisted user. Then these users are recommended in the order of their scores.

The preferences of a prospective bride is as follows: age - 24 to 26, height - 5.9 to 6, religion - Hindu, caste - Arya, education - Diploma, occupation - Government, annual income - 8 lakhs to 10 lakhs., body type - Average, complexion - Wheatish, Diet - Vegetarian, Smoke - No habits, Drink - No habits. The profiles in Table 3.1 are active bride grooms in the system.

Table 3.1: Profiles of active bride grooms

U-ID	Age	Hei	Rlgn	Caste	Edu	Occup	Inc	Body	Cmlx	Diet	Smo	Drnk
21	26	5.7	Hindu	Arya	Masters	Govt	4L	Athl	Dark	Veg	No	No
22	24	5.5	Hindu	Arya	Doctors	Govt	7L	Slim	Dark	Veg	No	No
23	30	5.1	Hindu	Arya	Doctors	Govt	14L	Avg	Whtsh	Veg	No	No
24	22	4.10	Hindu	Arya	HSC	Govt	8L	Avg	Fair	Veg	No	No
25	26	5.2	Hindu	Arya	Bchlr	Govt	3L	Hvy	VFair	Veg	No	No

If we run the algorithm on this data, the recommendations for active bride are shown in

Table 3.2. For this sample data, we assumed that the weight of each attribute is 10.

Table 3.2: Recommendations for Active Bride

User-ID	Score
21	49.3
24	48.4
22	46.45
23	40.2
25	44.65

If the same user has already rated 140 profiles positively and out of those 140 users if 10 are in age-group(AG) 1, 10 in AG 2, 20 in AG 3, 30 in AG 4, 20 in AG 5, 10 in AG 6 and 20 each in AG 7 and 8. Similarly, if the distribution for *Height* attribute is [10,10,10,30,30,10,10,30], for *Education* attribute is [10,20,50,30,20,10], for *Annual Income* is [10,10,10,30,10,10,30,30], for *Body Type* is [20,20,40,60] and for *Complexion* is [20,40,50,30], then the recommendations for the bride would be different as shown in Table 3.3. The system learns user's implicit preferences from his/her ratings and gives recommendations accordingly.

Table 3.3: Advanced Recommendations for Active Bride

User-ID	Score
25	52.7
22	43.06
21	41.73
24	40.06
23	38.06

3.1.2 RESULTS FOR SCORING ALGORITHM

Upon applying the algorithm on the stated User-Bride data according to User-Bridegroom preferences,

Firstly algorithms results in the following order (Algorithm checks only Most Preferred (MP)):

$$3 > 4 > 6 > 5 > 7 > 2$$

If the user is not satisfied with the prior recommendations and ready to receive more recommendations (Algorithm checks the best among the leftover recommended Users (MP + Best (P))):

$$3 > 4 > 7 > 5 > 2 > 6$$

If the user is further does not get satisfied with the recommendations and ready to concede his Most Preferred preferences and want to check further more recommendations (MP + Better (P) + Best(LP)):

$$3 > 4 > 7 > 2 > 5 > 6$$

If the user is ready to look over the whole database, then the algorithm is ready to display the recommendations according to the User's perspective (MP + Almost(P) + Better(LP)) :

$$3 > 4 > 7 > 2 > 5 > 6$$

	Educational Qualification	Occupation	Earnings	Family status	Location	Age	Height	Complexion	Body type	Diet
M P	M.com	Software	6,7	Rich class	Metro	21,22,23,24, 25	5'7,5'8, 5'9	Very Fair, Fair	Slim	Vegetarian
P	Chartered Accountant	Businessman	4,5	Upper middle class	Urban, Semi Urban	19,20,26,27	Anything	Whitish, Whitish Brown	Athletic	Eggitarian
L P	Anything	Anything	0,1,2,3	Middle class	Rural	18,28, 29	Anything	Dark	Heavy	Non Vegetarian

Figure 3.1: User-Bridegroom Preferential List

	Educational Qualification	Occupation	Earnings	Family status	Location	Age	Height	Complexion	Body	Diet
2	B.E	Banking	4	Middle class	Rural	24	5'7	Fair	Slim	Eggitarian
3	MBA	Logistics	6	Rich class	Urban	27	5'3	Very Fair	Heavy	Vegetarian
4	MCA	Software	4	Middle class	Semi Urban	26	5'2	Whitish Brown	Slim	Vegetarian
5	MD/MS	Teaching	7	Middle class	Semi Urban	29	5'2	Dark	Athletic	Eggitarian
6	BAMS	Looking for a job	0	Middle class	Rural	25	5'8	Very Fair	Slim	Vegetarian
7	Chartered accountant	ML/LLM	0	Middle class	Rural	25	5'8	Very Fair	Slim	Non Vegetarian

Figure 3.2: User-Bride Profile

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1 CONCLUSION

Recommender systems have always been powerful as means of new technology that is used for proving to be an essence in additional value for a business from its user databases. These systems help users find a commodity or an tradeable good that can be easily obtained from a business. Recommender systems benefit users by enabling them to find items they like. On contrary, these recommender systems help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web. New algorithms are of much need so as to dramatically improve the scalability of recommender systems.

This paper focuses on explaining the vital importance of improvisation over the existing recommender systems, which were much involved in getting more information from the user for knowing the partner preference that were much far away from providing perfect matches to their users. Our partially developed algorithm have shown that proposed recommender systems not only gives a new way of gaining information from users but also explains the mutual acceptance by providing matches with much similarity between them and not only are we providing N-recommendations but the order of each match that is provided to active user is taken into account, its an improvement in the field known less important.

4.2 FUTURE WORK

The following algorithm is partially developed and has the scope of improvement in the following areas:

- 1) The *Dataset* - As this project requires a large dataset which is eventually obtained by big companies that is not readily available to use is the major drawback so we choose to test having a small sample of the data.
- 2) Like-factor is to be added. Level 5 recommendation is to be introduced where α , β are values strictly dependent on number of most preferred profiles, preferred profiles and finally least preferred profiles.
- 3) Weights of attributes can be changed based on user onsite behaviour.

REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng*, 17:734 – 749, 2005.
- [2] B. Shapira F. Ricci, L. Rokach. *Recommender Systems Handbook*. Springer, 2011.
- [3] J. Riedl M. D. Ekstrand and J. A. Konstan. Collaborative filtering recommender systems, foundations and trends in human-computer interaction. 4:175–243, 2011.
- [4] T. Chung I. Koprinska L. A. S. Pizzato, T. Rej and J. Kay. Recon: a reciprocal recommender for online dating. 1:207 – 214, 2010.
- [5] J. A. Konstan and J. Riedl. Recommender systems: from algorithms to user experience. *User Model. User-Adapt. Interact*.
- [6] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst*, 22(1):143 – 177, 2004.
- [7] L. Brozovsky and V. Petricek. Recommender system for online dating service in proceedings of znalosti. *2007 Conference, Ostrava*, 2007.
- [8] L. Li and T. Li. *Meet: a generalized framework for reciprocal recommender systems*. CIKM, 2012.
- [9] K. Yacef L. Pizzato J. Kay J. Akehurst, I. Koprinska and T. Rej. A hybrid content-collaborative reciprocal recommender for online dating.
- [10] Kedar Joshi and Sushil Kumar. Matchmaking using fuzzy analytical hierarchy process, compatibility measure and stable matching for online matrimony in india. *Multi-Crit. Decis. Anal*, 19:56 – 66, 2012.
- [11] Mohana Priya.R Sampath.M.K, Nithya.C. Enhancing recommender system for matrimonial sites using collaborative filtering method. *IJCSEC*, 3:835–83, 2015.
- [12] Mrs.M.A.Potey Smita Khavate. Partial fuzzy recommender system for indian matrimony.