BORGO: A BOOK RECOMMENDER FOR READING GROUPS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAYIL SAÇAK DÜZGÜN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2012

Approval of the thesis:

BORGO: A BOOK RECOMMENDER FOR READING GROUPS

submitted by **SAYIL SAÇAK DÜZGÜN** in partial fulfillment of the requirements for the degree of
Master of Science in Computer Engineering Department, Middle East Technical University by,

Prof. Dr. Canan ÖZGEN
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Adnan YAZICI
Head of Department, **Computer Engineering** ________________

Dr. Ayşenur BİRTÜRK
Supervisor, **Computer Engineering** ________________

Examining Committee Members:

Prof. Dr. Nihan Kesim Çiçekli
Computer Engineering, METU ________________

Dr. Ayşenur Birtürk
Computer Engineering, METU ________________

Assoc. Prof. Pınar Şenkul
Computer Engineering, METU ________________

Assoc. Prof. Tolga Can
Computer Engineering, METU ________________

Assoc. Prof. Sevgi Özkan
Information Systems, METU ________________

Date: ________________

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

Name, Last Name: SAYIL SAÇAK DÜZGÜN

Signature :

ABSTRACT

BORGO: A BOOK RECOMMENDER FOR READING GROUPS

DÜZGÜN, Sayıl SAÇAK
M.S., Department of Computer Engineering
Supervisor : Dr. Ayşenur BİRTÜRK

January 2012, 87 pages

With the increasing amount of data on web, people start to need tools which will help them to deal with the most significant ones among the thousands. The idea of a system which recommends items to its users emerged to fulfill this inevitable need. But most of the recommender systems make recommendations for individuals. On the other hand, some people need recommendation for items which they will use or for activities which they will attend together. Group recommenders serve for these purposes. Group recommenders diverge from individual recommenders such that they need to aggregate members of the group in a joint model, and in order to do so, they need a user satisfaction function. There are two different aggregation methods and a few different satisfaction functions for group recommendation process. Reading groups domain is a new domain for group recommenders. In this thesis we propose a web based group recommender system which is called BoRGo: Book Recommender for Reading Groups , for reading groups domain. BoRGo uses a new information filtering technique and present a media for post recommendation processes. We present comparative evaluation results of this new technique in this thesis.

Keywords: Web Personalization, Information Filtering, Recommender Systems, Group Recommenders

ÖZ

BORGO: GRUPLAR İÇİN KİTAP TAVSİYE SİSTEMİ

DÜZGÜN, Sayıl SAÇAK
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Dr. Ayşenur BİRTÜRK

Ocak 2012, 87 sayfa

Web üzerindeki veri miktarındaki artış ile birikte, insanlar binlercesi arasından kendileri için en alakalı verilerle uğraşmalarını sağlayacak araçlara ihtiyaç duymaya başladılar. Kaçınılmaz olan bu ihtiyacı karşılamak üzere kullanıcılarına tavsiyelerde bulunan bir sistem fikri ortaya çıktı. Fakat tavsiye sistemlerinin birçoğu sadece bireylere yönelik tavsiyede bulunuyorlar. Diğer taraftan bazı insanlar diğer insanlar ile birlikte kullanacakları şeyler ya da birlikte katılacakları aktiviteler için tavsiyeye ihtiyaç duyuyorlar. Grup tavsiye sistemleri bu amaca hizmet ederler. Grup tavsiye sistemleri bireysel tavsiye sistemlerinden şu noktalarda ayrılırlar; grup tavsiye sistemleri grup üyelerini bir modelde birleştirme ihtiyacı duyarlar ve bunu yapabilmek için de bir kullanıcı hoşnutluk fonksiyonuna ihtiyaç duyarlar. Grup tavsiye işlemi için iki farklı birleştirme yöntemi ve birkaç tane farklı hoşnutluk fonksiyonu bulunmakta. Okuma grupları alanı grup tavsiye sistemleri için yeni bir alan. Biz bu tez çalışmasında okuma grupları alanında, BoRGo: Okuma Grupları İçin Kitap Tavsiye Sistemi adında, web tabanlı bir grup tavsiye sistemi arz ediyoruz. BoRGo yeni bir bilgi filitreleme tekniği kullanmakta ve tavsiye sonrası süreç için bir ortam sunmaktadır. Tezimizde bu yeni tekniğe ait karşılaştırmalı değerlendirme sonuçlarını sunmaktayız.

Anahtar Kelimeler: WWW Kişiselleştirme, Tavsiye Sistemleri, İçerik Bazlı Filtreleme, Gruplar İçin Tavsiye Sistemleri

To my beloved husband and my precious family

ACKNOWLEDGMENTS

First of all I would like to thank my supervisor Dr. Ayşenur Birtürk for her guidance and support throughout this thesis work.

My special thanks to my mother and father for their lifelong love and support and to my husband for everything he gave to me.

I would like to thank my colleagues in METU CC and all of my friends whom gave their support to me.

I also would like to express gratitude to the members of my jury, for reviewing this thesis.

I want to thank to the anonymous reviewers of ACM RecSys Decisions'11 for their valuable comments.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS x

LIST OF TABLES xiii

LIST OF FIGURES xv

CHAPTERS

1 INTRODUCTION 1

2 BACKGROUND INFORMATION 4

2.1 Recommendation Problem 4

2.2 Recommendation Techniques 6

2.2.1 Content Based Recommendation Systems 6

2.2.2 Collaborative Filtering Systems 8

2.2.3 Hybrid Systems 9

2.2.4 Semantic Recommender Systems 12

2.3 Group Recommendation Problem 12

2.3.1 Multi-Dimensional Recommender Systems 12

2.3.2 Group Recommender Systems 14

2.3.2.1 Aggregation Methods 16

2.3.2.2 Satisfaction Functions 21

2.3.2.3 Mediation Techniques 27

2.3.2.4 Domain Based Analysis of Group Recommender Systems 29

3 BoRGo: BOOK RECOMMENDER FOR READING GROUPS 32
3.1 Motivation 32
3.2 Domain Information 33
3.3 System Overview 34
3.4 System Architecture And Implementation Details 37
3.4.1 Individual Recommender 37
3.4.1.1 User Profile 38
3.4.1.2 Book Profile 41
3.4.1.3 Individual Recommendation Process And Filtering Method 44
First Filtering Method Used in BoRGo . . 44
Second Filtering Method Used in BoRGo 45
3.4.2 Group Recommender 46
3.4.2.1 Aggregator 47
Example 48
3.4.2.2 Mediator 52
4 EVALUATION OF THE SYSTEM 57
4.1 Data Set 57
4.2 Evaluation Metrics 60
4.3 Experiments With Using First Filtering Method 62
4.4 Experiments With Using Second Filtering Method 63
4.4.1 Evaluation of BoRGo with Different Sized Groups . . 65
4.4.2 Evaluation of BoRGo with the Groups Whose Ratings Are Dense 66
4.4.3 User Studies 67
4.5 User Studies For After Recommendation Process 67
5 CONCLUSION AND FUTURE WORK 79
5.1 Conclusion 79
REFERENCES 82
APPENDICES

A PUBLICATIONS 87

LIST OF TABLES

TABLES

Table 2.1 Item user rating matrix 5

Table 2.2 After recommendation process item user rating matrix 6

Table 2.3 Recommendation techniques 10

Table 2.4 Hybridization methods 11

Table 2.5 Ratings for the group 23

Table 2.6 Results of Plurality Voting strategy 23

Table 2.7 Results of Average strategy 24

Table 2.8 Results of Borda Count strategy 24

Table 2.9 Results of Copeland Rule strategy 25

Table 2.10 Results of Approval Voting strategy 25

Table 2.11 Results of Least Misery strategy 26

Table 2.12 Results of Most Pleasure strategy 26

Table 2.13 Results of Average Without Misery strategy 27

Table 2.14 List of group recommenders according to their domains 30

Table 3.1 Individual recommendation list for the Member1 49

Table 3.2 Individual recommendation list for the Member2 50

Table 3.3 Individual recommendation list for the Member3 50

Table 3.4 Group recommendation scores which are calculated for all of the books in individual recommendation lists of group members. RS indicates Recommendation Score 51

Table 3.5 Group recommendation list for the example reading group 52

Table 4.1 Alphabetically grouped List of dimensions in book dataset 58
Table 4.2 Rating counts of users 59
Table 4.3 Categorization of items 61
Table 4.4 Evaluation results of the experiments with first filtering method we used in BoRGo. 63
Table 4.5 Evaluation results of the experiments with second filtering method we used in BoRGo. 64
Table 4.6 Evaluation results of different sized random groups. 65
Table 4.7 Evaluation results of groups with dense ratings. 66
Table 4.8 Recommendation list of first group and member responses to the items. 69
Table 4.9 Recommendation list of first group after partial recommendation. . . 71
Table 4.10 Recommendation list of second group and member responses to the items. 73
Table 4.11 Recommendation list of third group and member responses to the items. 75
Table 4.12 Recommendation list of third group after partial recommendation. . 77

LIST OF FIGURES

FIGURES

Figure 2.1 Joint Recommendation List Method 18

Figure 2.2 Joint User Model Method 21

Figure 3.1 Flow chart of BoRGo 35

Figure 3.2 Web page organization of BoRGo 36

Figure 3.3 Rating history of the user 37

Figure 3.4 Users table 39

Figure 3.5 BookRatings table 40

Figure 3.6 UserFeatures table 40

Figure 3.7 Books table 42

Figure 3.8 Dimensions table 42

Figure 3.9 Features table 43

Figure 3.10 BookFeatures table 44

Figure 3.11 Calculate Recommendation Score algorithm 45

Figure 3.12 Calculate Recommendation Score 2 algorithm 46

Figure 3.13 Groups table 47

Figure 3.14 GroupUsers table 47

Figure 3.15 Edit Group Recommendation screen 53

Figure 3.16 ”Member Preferences” screen 54

Figure 3.17 Finding negatively rated books 56

Figure 4.1 Two examples from the book dataset. 59

CHAPTER 1

INTRODUCTION

Internet is the dominant way of the accessing information about anything nowadays. People use e-commerce sites like amazon [1] or e-bay [2] for shopping, web sites like youtube [3] to watch videos. Only the number of mp3s which can be downloaded from amazon [1] is 17 million. When you search for books via the same site 32,652,437 results are found. e-bay [2] provides 114,075 music DVD just in ”world music” subsection. When you simply type the word ”comedy” to the search bar of the youtube [3], you are getting lost among 1,790,000 videos. When you type the word ”e-commerce” to the popular search engine google [4], it founds 39.900.000 different sites for you. People can get lost easily among this information overload. It is really hard to find the video which you really want to see or download the songs which you really enjoy. It is clear that people need some tools which help them to choose wright items. Those tools are called ”Recommender Systems” in literature.

Recommender systems are hot research topics since mids of 1990s [5–7]. Research interest in this area is still alive in the second decade of new millennium because the more usage of the web increases the more the need of such tools deepens. When we look back to the roots of recommender systems, early researches in cognitive science, approximation theory, information retrieval, forecasting theories, management science, consumer choice modeling in marketing [8–13] can be seen as the base and inspiring works of the recommender systems.

There are two main recommendation approaches in recommendation system literature [14].First approach is collaborative filtering method and the second one is content based filtering method. Also some hybrid methods are used in recommender systems [14]. But nowadays semantic recommender systems [15] are very popular as third approach in recommendation research area.

Most of the current recommender systems recommend items only to individuals but in real life people often use items or do activities with groups of people in domains like movies, vacations, tourist attractions, music, and restaurants. Thus recommenders which aim groups of people are needed. Group recommenders diverge from individual recommenders such that they need to aggregate members of the group in a joint model because of the fact that individuals with different profiles composes the group itself, and in order to do this aggregation they need a user satisfaction function. Members of a group may need to get help to decide on the items of recommendation list therefore a group recommender should also provide a media to groups which they can use to discuss and decide what to do with the recommended list.

In this study we propose a content based book recommender for reading groups which is called BoRGo (Book Recommender for Reading Groups). BoRGo is designed to give recommendations to groups as well as to individuals by using book, user and group profiles. BoRGo is also capable of helping reading groups to determine the final book lists which will be read throughout the year. For example members of the reading group in our university first choose books to read and after reading them, they hold monthly discussion sessions. Choosing the books which most members would find interesting can be difficult. It is particularly hard to arrange meetings during the summer holidays, but they need to do this in order to select the books which they will read in the following year. As far as we know, BoRGo is the first study of group recommenders in reading group domain and the first web application which social communities can use to decide on the books to be read via internet without organizing face to face meetings.

This thesis consists of 5 chapters. The remaining 4 chapters are organized as follows.

In Chapter 2, the main topics in recommendation systems literature are presented. The recommendation problem is formalized, main recommendation approaches along with the limitations are explained, and then group recommendation problem , aggregation methods, satisfaction functions and mediation techniques are mentioned.

In Chapter 3, the proposed group recommender is explained in detail. First, reading group domain is explained. After that, proposed approach is overviewed. Then system architecture and system components are explained. At the end of the chapter implementation details of filtering method, aggregation method and satisfaction function are given.

In Chapter 4, details of the dataset are mentioned. After that evaluation metrics which are used are explained. Then the details of the three evaluation phases are presented and results are discussed.

In Chapter 5, conclusion is presented and future work of the proposed work is discussed.

CHAPTER 2

BACKGROUND INFORMATION

2.1 Recommendation Problem

People use recommendations in every part of their daily lives. For example they read the comments of others about movies or simply ask for their friends' opinions before they go to cinema. Women ask their friends for good recipes before they host guests for dinner. In the world of today a main source of information is internet but there are numerous web sites containing thousand or even millions of items to choose. Without help it is nearly impossible to narrow down the list of items which is relevant to people.

Recommendation problem can be defined as finding most relevant items for individuals by using some related background information about that individual. Domain of the recommendations can vary from books to music, movies to recipes .

Recommenders Systems are computer based techniques which are used to help users in solution of different recommendation problems. According to [14] it can be said that the Grundy [15] system is the first example of the recommender systems. the Grundy system uses stereotypes as a mechanism to build user models and using the user models it recommends relevant books to each user.

Recommendation problem can be defined formally as follows [14]. Let U be the set of possible users and I be the set of possible items for example all of the possible books for a book recommender. Let s be a satisfaction function which shows the satisfaction which user u gets from an item i. s: U x I → R, where R represents a range of integers. Then we can say that recommendation problem is

$$\forall u \in U, i \in I, i'_{\mathrm{u}} = argmaxs(u, i) \tag{2.1}$$

Generally in recommender systems satisfaction is represented by the rating of the user to the item. Therefore we can say that basically recommendation problem is to calculate the estimated rating of a user for an item that he/she haven't rated yet by using a satisfaction function.

In Recommendation systems, ratings are represented by item user rating matrices. When the count of users and items are big usually rating matrices become sparse. An example of item user rating matrix can be seen in (Table 2.1). Rating range for this example is between 0 and 10. In this matrix columns represent users and rows represent items. Empty cells correspond to the items which are not rated by the user in related column. These unknown ratings predicted by the recommender system and according to those predicted ratings items are recommended to the users.

Table 2.1: Item user rating matrix

	B1	B2	B3	B4	B5	B6	B7	B8	B9	..	Bn
U1			7								
U2	8							3			
U3					5			6			
..											
Un				9							

Unknown ratings are predicted by the recommender system and according to predicted ratings items are recommended to the users after recommendation process. For example after recommendation process same matrix in (Table 2.1) turns into the matrix in (Table 2.2) with the predicted ratings.

Table 2.2: After recommendation process item user rating matrix

	B1	B2	B3	B4	B5	B6	B7	B8	B9	..	Bn
U1	4	6	7	10	5	2	6	1	3	..	9
U2	8	7	8	4	10	3	2	3	5	..	8
U3	1	5	4	9	5	8	7	6	4	..	4
..	..	..	..	..	..	..	..	..	..	..	..
Un	7	3	5	9	8	3	6	2	4	..	10

2.2 Recommendation Techniques

There are two traditional approaches used in recommenders systems literature [14]. Some different hybrid techniques emerged in need of overcoming weaknesses of these two methods in time [14] [16]. And as a new approach semantic recommendation [17] is used in some relatively new studies in literature. In this section these four approaches are examined. There are also other recommendation techniques like demographic, utility based and knowledge based recommendation techniques [16]. These techniques are not examined in this thesis but mentioned briefly in Hybrid Systems subsection.

2.2.1 Content Based Recommendation Systems

Content based recommender systems uses content information of the items which have been rated by the user. For example a content based book recommender first analysis commonalities of books the user has already rated then recommend the books which are most appropriate for the user's tastes.

In content based recommendation both users and items have profiles. User profile represents tastes and preferences of the user. Necessary information to profile a user can be gathered from user explicitly or from transactional behavior implicitly. [14] Item profiles consist of features and weights of them which are called feature weight vectors and are used for determining that if a item is suitable for a user.

In content based recommenders features which compose item profiles have feature scores which identifies the importance of each feature. For example for a content based book recommender some features are more important than others to distinguish a book.

Content based recommendation rooted to Information Retrieval therefore the measures which are used to calculate weight of a feature inherited from Information Retrieval [14] . [21] mentions some different measures to decide weight of features;

- Number of Occurrence: weight of the word equals to frequency of the word in the sentence.
- Binary Representation of Number of Occurrence: Weight of the word equals to 1 if the word is seen in the sentence, otherwise equals to 0.
- TF-IDF (Term Frequency-Inverse Document Frequency): Weight of the word equals to TF-IDF value of the word. ”The importance of a word is high if it is frequent in the sentence, but less frequent in the document” [21].
- Log Entropy: Weight of the word equals to log entropy value of the word.
- Root Type: If the root type of the word is noun, Weight of the word equals to frequency of the word in the sentence otherwise weight of the word equals to 0.

There are examples of content based recommenders which use different methods like Bayesian classifiers, clustering , decision trees, artificial neural networks [22, 23] instead of using traditional information retrieval based heuristics. These approaches use a model which is constructed by using statistical and machine learning techniques to recommend items.

There are some limitations of content based recommenders these are;

- New user problem
- Limited content analysis
- Overspecialization

Content based recommenders can not give trusted recommendation unless user rates a sufficient number of items. Because they can not understand and construct user profiles with a few ratings correctly. This is called new user problem. Also some items are not suitable to represented by features like images, audio and video streams. If the two different items have exactly the same feature set they can not be distinguished by the recommender. These two problems are limited content analysis problems. Content based recommenders recommend similar items to a particular user and this causes overspecialization. For example a content based recipe recommender would not recommend Turkish meals to a user who has not rated any Turkish meal or would recommend only the French meals to a user who liked a French meal.

2.2.2 Collaborative Filtering Systems

Collaborative recommender systems [14] try to find possible satisfaction of items for a particular user by using the ratings given by other users. In order to do that recommender first finds the similar users;f or example, users who rated same books similarly. Then recommender finds the books which these similar users mostly liked and recommends them. It can be claimed that the Tapestry [18] system is the first example of collaborative recommenders. In Tapestry system users must select the similar users explicitly.

Collaborative systems are divided into two groups according to their algorithms [19]: Memory Based and Model Based. Memory Based collaborative recommenders make recommendations based on all of the previously rated items by the users. On the other hand Model Based collaborative recommenders use previous ratings to elicit a model by using some statistical and machine learning techniques; then use the model to make recommendations.

Collaborative recommender systems have some limitations: [14].

- New User Problem
- New Item Problem
- Sparseness

Like content based systems , collaborative filtering systems need to learn preferences of the new users. On the other hand new items cause problems in collaborative filtering systems. Because of the fact that collaborative recommender systems use other users ratings to recommend an item to users, until a specific number of users rates the new item , recommender can not recommend that item to users. Generally user item rating matrices are sparse in recommender systems. User and item datasets are huge and intersections of columns and rows of the matrix are usually empty. It is hard for the recommender system to give recommendation to marginal users because it is hard to find similar users to them. Also items which are rarely rated can not be recommended to the users who like new items. These are side effects of sparse item user matrices.

2.2.3 Hybrid Systems

Hybrid systems can be an efficient approach, in order to overcome limitations of previously mentioned techniques. By combining multiple recommendation techniques performance of a recommender system can be improved [16]. (Table 2.3) [16] lists different recommendation techniques and (Table 2.4) [16] lists hybridization methods which can be used with these techniques.

Table 2.3: Recommendation techniques

Technique	Background	Input	Process
Collaborative	Ratings from U of items in I.	Ratings from u of items in I.	Identify users in U similar to u, and extrapolate from their ratings of i.
Content-based	Features of items in I.	u' ratings of items in I.	Generate a classifier that fits u's rating behavior and use it on i.
Demographic	Demographic information about U and their ratings of items in I.	Demographic information about u.	Identify users that are demographically similar to u, and extrapolate from their ratings of i.
Utility-based	Features of items in I.	A utility function over items in I that describes u's preferences.	Apply the function to the items and determine i's rank.
Knowledge-based	Features of items in I. knowledge of how these items meet a user's needs.	a description of u's needs or interests.	Infer a match between i and u's need.

Table 2.4: Hybridization methods

Method	Description
Weighted	The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation.
Switching	The system switches between recommendation techniques depending on the 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 recommender refines the recommendation given by another.
Feature augmentation	Output from one technique is used as an input feature to another.
Meta-lavel	The model learned by one recommender is used as an input to another.

Although it is possible to hybridize all types of recommendation techniques, mostly content based and collaborative filtering techniques are used in hybrid recommenders. [14] lists different ways to use of these two techniques in hybrid recommenders:

- combining predictions of separately implemented techniques
- including some characteristics of content based technique in collaborative filtering approach
- including some characteristics of collaborative technique in content base approach
- building a unifying new model which includes both of some characteristics of collaborative technique and content base technique

2.2.4 Semantic Recommender Systems

Semantic Recommender Systems can be seen as a sub type of knowledge based recommenders because they use a knowledge base to operate. They use Semantic Web technologies to give recommendation. [17]. Semantic Recommender Systems can be used to overcome sparseness and new item problems [20] but they need knowledge engineering activities [16]. Types of semantic recommender systems are below [17]:

- Vocabulary or ontology based systems
- Trust Network based systems
- Context adaptable systems

2.3 Group Recommendation Problem

Recommender systems have been used in different domains with different approaches like mentioned in previous sections. However, to be used in solution of more complex recommendation problems all of those methods need to be extended and improved [14] [24]. Some of recommendation problems need a different aspect of recommendation and a recommender must take different factors into consideration for solution of these problems.

2.3.1 Multi-Dimensional Recommender Systems

Most of current recommender systems produce recommendation in only two dimensional User x Item space. But in reality people decide whether they like an item or not depending on with whom they use that item or when they use that item etc. Therefore recommender systems also need to take other dimensions of information into consideration.

[14] [24] [25] propose multi-dimensional model for solution of more complex recommendation problems. In traditional recommender systems recommendation problem can be defined as follows;

$$R : User \times Item \rightarrow Rating \tag{2.2}$$

For multi-dimensional problems definition of the problem changes into following ;

$$R : Dimension_1 \times Dimension_2 \times ..... \times Dimension_n \rightarrow Rating \tag{2.3}$$

Think of a multi-dimensional recommender system which recommends hotels to users with following dimensions;

- Hotel: all hotels which can be recommended by the application.
- Person: all users which can get recommendation from the application.
- Companion: person or group of people with whom the user who need recommendation will stay in the hotel.

For this example ratings of a user to a hotel would change according to companions of the user. For example an user 20 years old user A may prefer a hotel which has a disco more if he would stay in that hotel with his same aged friends. But A may prefer a hotel with a good breakfast and dinner facility if he would stay in that hotel with his colleagues. In this case rating function would be;

$$R : Hotel \times Person \times Companion \rightarrow Rating \tag{2.4}$$

Dimensionality of this example can be increased with following dimensions;

- Time: when the user would stay at the hotel
- Location: whereabouts of the hotel which the user would stay

When the previous example reconsidered with these new dimensions, situation may turn into that: User A may prefer a country hotel which has an open disco if he would stay at that hotel with his same aged friends in summer. But A may prefer a city centered hotel with a big indoor restaurant if he would stay at that hotel with his colleagues in winter. With this new dimension rating function would be;

$$R : Hotel \times Person \times Companion \times Time \times Location \rightarrow Rating \tag{2.5}$$

To decide to dimensionality level is a problem for multi-dimensional recommenders. [25] suggests that this problem is related to feature selection problem in data mining [26] and statistics [27] area. When feature values of dimensions D1, D2, ... Dn-1 remain same and only feature values of Dn changes, if any feature value of Dn does not affect the rating in rating function, it means nth dimension is not needed and can be removed from recommendation mechanism. For example assume that in the previous example if possible location values like city centered, country side etc. does not affect the distribution of the ratings when the feature values for other dimensions remains same for each location value. In that case location value is not needed as a dimension in recommendation process.

Multi-dimensional recommender systems need to aggregate measurements of different dimensions. [25] used a multidimensional database which support aggregation hierarchies for different dimensions and has capabilities to aggregate the measurements at different levels of the hierarchy [28, 29]

2.3.2 Group Recommender Systems

Most of the recommender systems are designed to produce recommendation to only individuals. But people generally do some activities like watching movies or going vacations, eating meal not by themselves but with group of people. Not only activities but also some items are consumed by groups. Group Recommender Systems are appropriate solutions of recommendation problem for groups of people. [31] and MusicFx [30] are the first examples of group recommender systems. [31] recommends videos to either individuals or groups by using collaborative filtering.

MusicFx [30] selects playing radio station in a sport center according to aggregation of preferences of members of the center who are present at the time.

Group Recommender Systems can be seen as a sub kind of Multi-Dimensional Recommender Systems which has three dimensions;

- Item: all items which can be recommended by the application.
- Person: all users which can get recommendation from the application.
- Companion: person or group of people with whom the user who need recommendation will consume the item.

And rating function for a Group Recommender System is;

$$R : Item \times Person \times Companion \rightarrow Rating \tag{2.6}$$

In a group recommender Person can be defined as the member of the group which asks for recommendation on behalf of the whole group. He/she can be anyone among the group. Companion can be defined as the remaining members of the group except the one which plays the role of Person. Item can be any activity or item which the Recommender System recommends to its users.

Group Recommender Systems aims to satisfy not only the individual person whom asks for the recommendation on behalf of the group but also the whole group members up to a certain degree. At this point different satisfaction functions can be chosen to decide the level of satisfaction of the group. Group Recommender Systems need to aggregate individual user preferences or models to produce a recommendation list for the group itself. These two concepts, Aggregation Methods and Satisfaction Functions, will be mentioned following sections of this chapter.

Groups may need to discuss and accept or not accept some of the items which are recommended by the Group Recommender System. And in order to that they need a media. This concept is one of the topics of group recommenders which are not addressed by many researches. The Mediation Techniques subsection addresses a few studies about this topic.

2.3.2.1 Aggregation Methods

Group recommenders need to aggregate individual preferences of all of the group members in order to recommend item for groups. Therefore they need a mechanism for aggregation. It can be seen from previous works [32] that there are two main methods for aggregation process.

- Joint recommendation list: In this method individual recommendation lists are produced for every member. Then these recommendation lists combined into a single joint list.
- Joint user model: In this method individual preferences or profiles aggregated before recommendation process and a single recommendation list generated for that joint model.

For producing a Joint Recommendation List first individual recommendation lists of every member are produced then these lists are combined in a joint recommendation list for the group by using a satisfaction function. Figure 1 shows recommendation process whit joint recommendation list method. This method has a powerful explanation advantage. As the individual lists are present it is easy to explain presence and order of the items in the final list. Individual recommendation lists also can be used in after recommendation process to help the group to come a consensus over the chosen items from the recommendation list. But this advantage can turn into a disadvantage easily in security and privacy perspective. Some users may not be pleased to share their individual lists with the rest of the group. In addition it can be time consuming for large scale groups because of creation of individual recommendation lists for each user in the group.

PolyLens [32] uses this method to aggregate individual member preferences while recommending movies to groups of people.

PocketRestaurantFinder [33] is one of the early examples of group recommenders. It recommends restaurants to groups of people according to their preferences and location. PocketRestaurantFinder [33] uses Joint Recommendation List method. It first calculates individual preference for each restaurant then sums up individual preferences to group preference and sorts restaurants according to these values.

INTRIGUE [34] is a group recommender system which recommends tourist attractions to groups. Groups are divided into homogenous subgroups by the user according to their demographic information and preferences. System first calculates satisfaction values for each subgroup then combines the subgroup-related satisfaction scores in a weighted way.

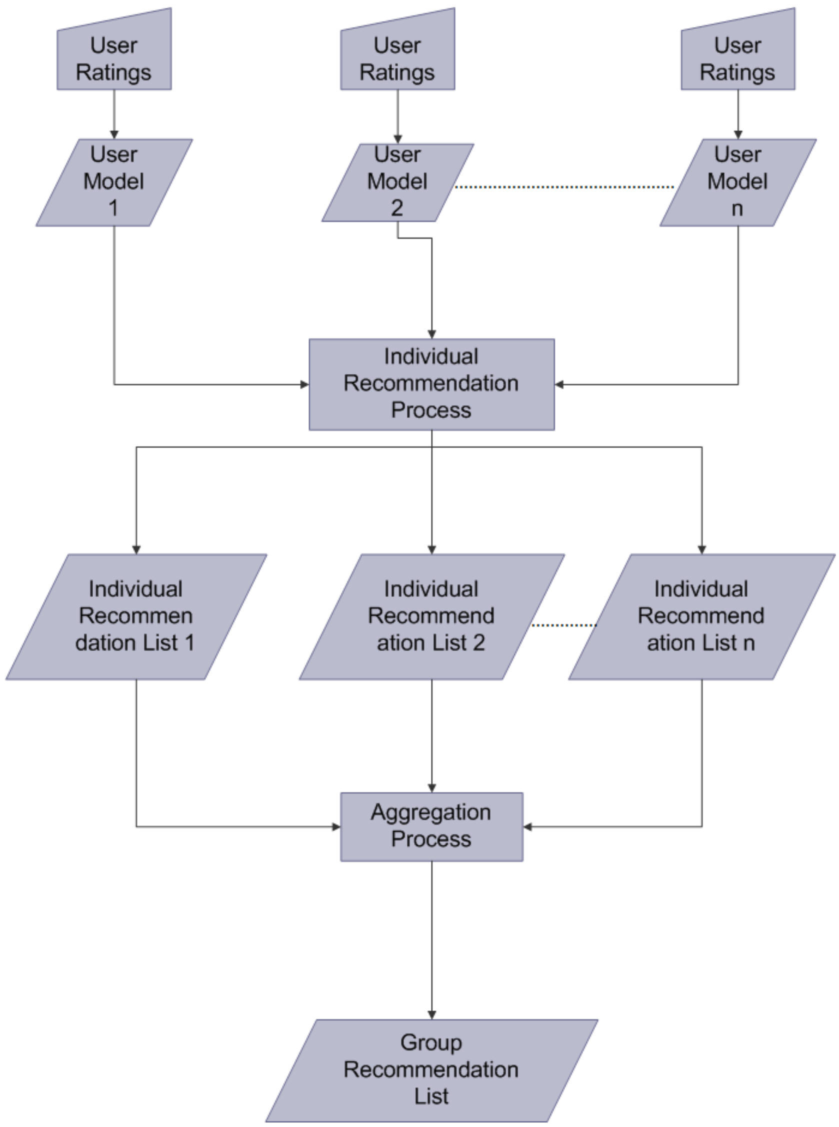


Figure 2.1: Joint Recommendation List Method

Group Recommender systems which are using joint user model aggregate individual user profiles by merging them into a common user profile for the group and then recommends a single list for this joint profile.Figure 2 shows recommendation process whit joint recommendation list method.

Merge process either can be done automatically by the recommender or manually by group members -or by the moderator of the group on behalf of group members. Manual merge process allows members of the group to explicitly predict their preferences and come a consensus on the joint user model. But it can be very time consuming. On the other hand automatic merge process is very practical compared to manual method. But in both methods it is harder to explain final recommendation list to the members compared to Joint Recommendation list method because this method can recommend some items which would not be seen in individual recommendation lists of the members of the group. This characteristic of Joint User Model method can also be seen as an advantage. Because it is sometimes good for a recommender to recommend marginal items to its users for novelty issues.

Travel Decision Form [35] uses manual merging method. It aims to make members of the group agree on a single way of selecting preferences of the group. In order to do that system gives a media to group members which they can learn each other's preferences which each of the members are represented by animated characters.

[36] recommends media to passengers in a vehicle. They use "total distance minimization" in automatic merge process of individual user profiles. Joint User Model is a suitable method for the groups which are not formed intentionally and are random like passengers in a vehicle. Because people who do not know each other do not care for each other's preferences as much as the people who are intentionally form a group like a group of people whom want to watch a movie together.

CATS [37] recommends skiing vacation alternatives to a group which is consist of four people who are sitting around an interactive tabletop. CATS produces recommendation for both individuals and groups simultaneously. It uses user feedbacks about current recommendations to update explicit user and group profiles and recommendations for individuals and group. This approach is called critiquing based recommendation [37–39]. When a user gives a new critique about a hotel joint group profile as well as the user profile is updated.

[40] recommends TV programs for group of people. System automatically merges individual user profiles into a joint user profile in two steps. In first step recommender decides which features included in the common user profile based on total distance minimization method. In second step it assigns weights to the features which are included in common profile for the group.

GRec-Oc [41] recommends items to online communities. GRec-Oc merges individuals profiles into group profiles and make single recommendation list for the joint user profile of the group. It uses two phases to recommend items to users. In first phase it uses collaborative filtering method to find items to recommend to the group. In second phase it uses content based filtering to remove irrelavant items from recommendation list to improve individuals' satisfaction. Authors evaluated their systems with business-major graduate and undergraduate students. They grouped users according to their ages,majors, personal interests, and career objectives so that individuals sharing similar and related profiles were grouped together. They used books as recommendation items.

[42] produce a list of individual preferences according to demographic, content-based, likes-based filtering type of recommendations in order to produce set of three lists of individual preferences for members of the group. These individual preferences processed with methods like aggregation, intersection or incremental intersection, and then turned into a joint user model. Group recommendation list is produces for this joint model.

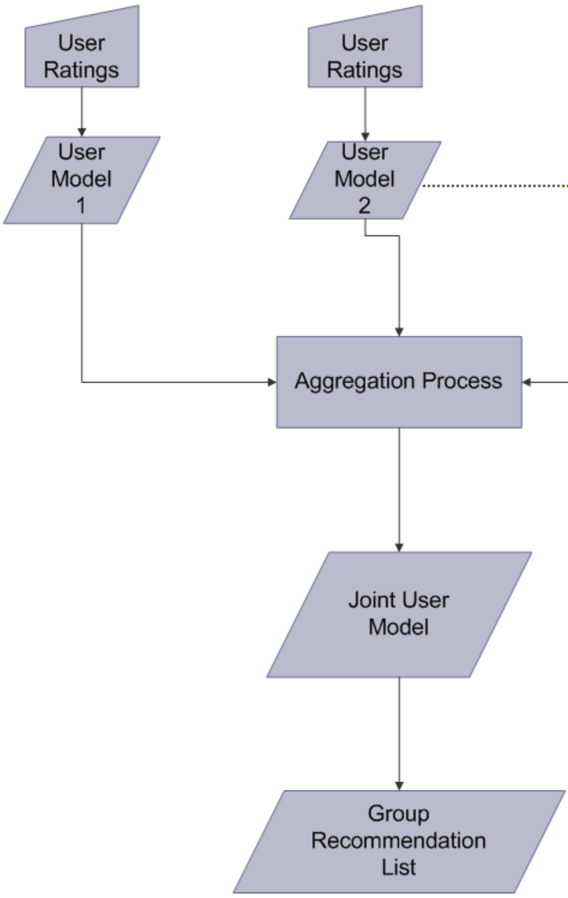


Figure 2.2: Joint User Model Method

2.3.2.2 Satisfaction Functions

Another main characteristic of group recommenders which separates them from individual recommenders that they need a satisfaction function during aggregation process both for joint recommendation list and joint user model methods. Satisfaction function determines what is best for the group according to individual choices and studied well in Economics, Politics, Sociology Math, MetaSearch, Database Middleware, Collaborative Filtering, and Multi-Agent systems [43–51]

[52] presents detailed research results on different satisfaction functions. It is known that rating behaviors of different users are also different. For example for user A 8 out of 10 means excellent because he never gives any item 9 or 10 but for user B it means normal because he gives 9 or 10 for the items he likes most. [52] assumes that all of the individuals have ”reasonable” rating behaviors. [52] suggests ten different

satisfaction functions;

- Plurality Voting: Each member of the group votes for her favorite item. Items are ordered from mostly voted to least voted ones.
- Average Strategy: Average of the ratings of individuals is calculated for each item and items are ordered according to these average rating values. INTRIGUE [34] uses this strategy as its satisfaction function.
- Borda Count [53]: points starting from 0 are appointed to the ordered items in individual lists. Then all individual points for each item are summed and items are ordered according to their group points for the group recommendation.
- Copeland Rule [54]: Alternatives are ordered according to the Copeland index: "the number of times an alternative beats other alternatives minus the number of times it loses to other alternatives"
- Approval Voting: Voters vote for all the items they wish. Items are ordered according to how much they voted.
- Least Misery Strategy: Group is as happy as the least happy member. PolyLens [32] uses this strategy. They assume that small groups of people watch movies together and it is meaningful to take into consideration of the least happy member's rating for small groups. The disadvantage of this strategy is that a minor opinion decides for the whole group's satisfaction. For example in a situation that remaining of the group wants to see a movie but only one person does not, that item would not be recommended to the group.
- Most Pleasure Strategy: Items are recommended according to their highest ratings.
- Average Without Misery Strategy: Average of the ratings of individuals are calculated for each item and items are ordered according to these average rating values but items which has a rating less than a threshold is simply disregarded. MusicFX [30] uses this strategy.
- Fairness Strategy: Top items are selected from each individual's list round by round. People may think that it is not really something bad to be recommended

the items you really hate as long as the items you love are also recommended.

- Most Respected Person Strategy: Ratings of the most respected person -or an expert- are taken as the ratings of the whole group. This strategy can be used for the groups of which one member is dominant like a group which is consists of an adult and some children.

Table 2.5: Ratings for the group

	B1	B2	B3	B4	B5	B6	B7	B8	B9
U1	5	6	7	10	6	2	5	4	9
U2	8	9	7	2	5	10	7	3	4
U3	9	7	1	9	7	10	8	6	10

(Table 2.5) shows example ratings for three member of a reading group for nine different books. All of the calculations of different satisfaction function use these ratings in fallowing part.

If Plurality Voting strategy is used as the satisfaction function ratings in first our U1 would vote for B4, U2 would vote for B6 and U3 would vote for B9 and B6. Since B6 gets two votes B6 wins first tour. In second tour U1 would vote for B4, U2 would vote for B2 and U3 would vote for B9. This tour would result in a tie and all of the three candidates would win. Sequence will be like in (Table 2.6);

Table 2.6: Results of Plurality Voting strategy

U1	B4	B4	B3	B3	B5	B8
U2	B6	B2	B1	B3, B7	B5	B8
U3	B6, B9	B9	B1	B7	B5	B8
Group	B6	B4, B2, B9	B1	B3, B7	B5	B8

Group list: B6, (B4, B2, B9), B1, (B3, B7), B5, B8

Average votes for each book would be like in (Table 2.7) if Average Strategy is used as the satisfaction function;

Table 2.7: Results of Average strategy

	B1	B2	B3	B4	B5	B6	B7	B8	B9
U1	5	6	7	10	6	2	5	4	9
U2	8	9	7	2	5	10	7	3	4
U3	9	7	1	9	7	10	8	6	10
Group	7	7	5	7	6	7	6	4	7

Group list: (B1, B2, B4, B6, B9), (B5, B7), B3, B8

If Borda Count would be used, for U1 B6 gets 0 point, B8 gets 1 point, B1 and B7 get both 2 points, B2 and B5 get both 3 points, B3 gets 4 points, B9 gets 5 points and B4 gets 6 points. (Table 2.8) shows points of all users for all of the books;

Table 2.8: Results of Borda Count strategy

	B1	B2	B3	B4	B5	B6	B7	B8	B9
U1	2	3	4	6	3	0	2	1	5
U2	5	6	4	0	3	7	4	1	2
U3	4	2	0	4	2	5	3	1	15
Group	11	11	8	10	8	12	9	3	12

Group list: (B6, B9), (B1, B2), B4, B7, (B3, B5), B8

Table (Table 2.9) shows results for the group if Copeland Rule would be applied to ratings. If B1 compared to B2 it is seen that U1 and U2 prefer B2 to B1, only U3 prefers B1 to B2. Thus B2 beats B1 according to Copeland index.

Table 2.9: Results of Copeland Rule strategy

	B1	B2	B3	B4	B5	B6	B7	B8	B9
B1	0	+	-	0	-	+	-	-	+
B2	-	0	-	+	-	+	-	-	+
B3	+	+	0	+	-	+	0	-	+
B4	0	-	-	0	-	+	-	-	+
B5	+	+	+	+	0	+	+	-	+
B6	-	-	-	-	-	0	-	-	0
B7	+	+	0	+	-	+	0	-	+
B8	+	+	+	+	+	+	+	0	+
B9	-	-	-	-	-	0	-	-	0
Group	+1	+2	-3	+3	-6	+7	-3	-8	+7

Group list: (B9,B6),B4,B2,B1,(B3,B7), B5,B8

For the Approval Voting if it is supposed that voters would like the items which they give above 6 U1 would vote for B3,B4 and B8. (Table 2.10) shows the votes of each user and for the group for Approval Voting;

Table 2.10: Results of Approval Voting strategy

	B1	B2	B3	B4	B5	B6	B7	B8	B9
U1			1	1					1
U2	1	1	1			1	1		
U3	1	1		1	1	1	1		1
Group	2	2	2	2	1	2	2	0	2

Group list: (B1, B2, B3, B4, B6, B7, B9), B5, B8

If the satisfaction function is chosen as the Least Misery strategy, least happy member's ratings would be group's ratings. (Table 2.11) shows results of Least Misery strategy;

Table 2.11: Results of Least Misery strategy

	B1	B2	B3	B4	B5	B6	B7	B8	B9
U1	5	6	7	10	6	2	5	4	9
U2	8	9	7	2	5	10	7	3	4
U3	9	7	1	9	7	10	8	6	10
Group	5	6	1	2	5	2	5	3	4

Group list: B2, (B1, B5, B7), B9, B8, (B4, B6), B3

If the satisfaction function is chosen as the Most Pleasure strategy, most happy member's ratings would be group's ratings. (Table 2.12) shows results of Most Pleasure strategy;

Table 2.12: Results of Most Pleasure strategy

	B1	B2	B3	B4	B5	B6	B7	B8	B9
U1	5	6	7	10	6	2	5	4	9
U2	8	9	7	2	5	10	7	3	4
U3	9	7	1	9	7	10	8	6	10
Group	9	9	7	10	7	10	8	6	10

Group list: (B4, B6, B9), (B1, B2), B7, (B3, B5), B8

If Average Without Misery (AWM) Strategy is used with a threshold -for example 4- B3,B4, B6 and B8 would be discarded. (Table 2.13) shows results of AWM strategy;

Table 2.13: Results of Average Without Misery strategy

	B1	B2	B3	B4	B5	B6	B7	B8	B9
U1	5	6	7	10	6	2	5	4	9
U2	8	9	7	2	5	10	7	3	4
U3	9	7	1	9	7	10	8	6	10
Group	7	7	-	-	6	-	6	-	7

Group list: (B1, B2, B9), (B5, B7)

If Fairness Strategy is used as the satisfaction function it can be inferred that U1 would choose B4, U2 would choose B6 and U3 would choose B9 in the first round. In second round U1 would choose B3 , U2 would choose B2 and U3 would choose B1. In third round U1 would choose B5, U2 would choose B7 and U3 would choose B8 finally. Group list would be like below;

Group list: B4, B6, B9, B3, B2, B1, B5, B7, B8

If the satisfaction function is chosen as Most Respected Person Strategy and U2 is the most respected person in the group then group list would be like below;

Group list: B6, B2, B1, (B3, B7), B5, B9, B8, B4

According to experiments done by [52]Average,Average Without Misery, and Least Misery strategies are most comprehensible ones for implementation. As it is mention before MusicFX [30], PolyLens [32], INTIGUE [34] use these strategies.

2.3.2.3 Mediation Techniques

Individual recommender system users need make a choice among the alternatives recommended by the system. Users need to understand why the recommender system make those recommendation for him while choosing one alternative over the others. Thus even in individual recommendation domain recommender systems should offer a media for after recommendation processes. [55]

For group recommendation after recommendation processes are obviously more complicated. Because for group recommendation it is not an individual decision which has to be done, a group consensus is needed. Especially when the group members does not know each other well and the group is formed intentionally but temporarily -like in Pocket Restaurant Finder [33] it can be hard to reach a consensus over the selection among the recommended alternatives. Even if the group members know each other well -like in PolyLens [32] they need to know what the others think about the possibilities.

[55] claims that if the members know each other's opinions about alternatives they can reconsider their choices, especially if they are made not very carefully, because members may care about equality and group solidarity. It is possible to share opinions directly if all of the group members are present and together when the recommender system gives recommendation. However if the group recommender is a web based application, group members would not be in the same place moreover they may even use the recommender system asynchronous.

[55] uses animated characters on behalf the members which can not communicate others syncrounsly to come a consensus over the joint user preference model. Currently logged in member of the group can see others preferences and ask for recommendation according to his preferences. At this level system shows a recommendation list based on his preferences, a recommendation list based on the current group preferences and list based on other members' current preferences to the member. The member can change his preferences to go along with the others' more at this point and ask for recommendation again and again until he is satisfied with his own preferences. After each member finishes specifying their preferences system goes into second phase. In second phase system displays a proposal to fill out preferences form for the whole group. Animated characters simulate the reaction of absent members of the group to the current proposal. Then system asks for the currently available member to respond to the proposal. At this point current member can accept or reject the proposal or he can change his preferences. System

continues to produce proposal until all of the members accept a proposal.

Group management interface of Pocket Restaurant Finder [33] allows users to send their preferences to others or receive others' preferences. But it does not offer a real solution for after recommendation processes.

PolyLens [32] adopts a composite recommendation interface approach which shows prediction for each member of the group as well as the group itself.

Maybe the only situation which it is not needed a after recommendation process is when the group is temporarily and unintentionally formed like people traveling in a vehicle in [36] or like people doing exercises in a sport center in MusicFx [30].

Yet after recommendation processes are essential for the success of a group recommender, this topic has not been handled by most of the group recommenders.

2.3.2.4 Domain Based Analysis of Group Recommender Systems

Recommenders systems serve users in many different domains. Some of them are commercial systems. There are relatively less group recommenders than individual ones. Group recommenders used in domains like music, restaurants, movies, vacations and sight seeing, multimedia, tv programs, books and recieps.

In this section we will examine group recommenders according to their domains. (Table 2.14) lists group recommenders according to their domains.

In music domain MusicFX [30] is a well-known example of the group recommenders. MusicFX selects the music playing in a sport center based on people's preferences that are currently in that sport center. MusicFX aggregates user preferences for the recommendation process.

PolyLens [32] and [73] are the examples of the group recommenders in movie domain. Both of them use recommendation list aggregation technique.

Table 2.14: List of group recommenders according to their domains

Domain	Group Recommender Systems
Music	MusicFX [30]
Restaurants	Pocket Restaurant Finder [33]
Movies	PolyLens [32]
Vacations and Sight Seeing	INTRIGUE [34] , Travel Decision Forum [55] , CATS [37]
Multimedia	An Adaptive In-Vehicle Multimedia Recommender [36]
TV Programs	TV4M [40]
Books	GRec-OC [41]

In restaurant domain, Pocket Restaurant Finder [33] is the example of group recommender systems which recommends restaurants to a group of people by taking the location of the group into account. System pools specified user preferences together and present a list of potential restaurants, sorted in order of expected desirability of the group.

In traveling and sight seeing domain there are more examples than other domains. INTRIGUE [34] is a group recommender system which offers sightseeing opportunities to demographically heterogeneous groups considering the time limits. Recommendation activity is based on a declarative representation of the knowledge about tourist attractions and on the application of fuzzy evaluation functions for ranking items. Travel Decision Forum [55] is a prototype which focuses on post recommendation process rather than on implementation. In the prototype, group members are trying to settle with their preferences which is called joint preference model. CATS [37] is designed to give recommendation to a group of friends planning to go a skiing vacation. It is a collaborative group recommender system. User feedbacks - also known as critiquing- are used to update explicit user models on a per-user-basis as well as a global user model.

An Adaptive In-Vehicle Multimedia Recommender [36] is the example of multimedia domain. System recommends multimedia items to travelling group of users. It aggregates passenger profiles, merges them in order to create a common profile and generates recommendations according to this common user profile.

Domain of TV4Mcite40 is tv program recommendation. It first merges user profiles, and then produces a single recommendation list according to the merged user profile.

GRec-OC [41] recommends scientific books to online communities. They first aggregate the preferences of members of the group and then make a single recommendation list for the group.

[74] is a good example of how group recommenders can be used in different domains. Their domain is recipes. They assign different weights to users in the groups by using four different weighting models and apply both of the aggregation methods and discuss the experimental results.

CHAPTER 3

BoRGo: BOOK RECOMMENDER FOR READING GROUPS

This chapter presents a content based book recommender for groups, BoRGo. Firstly reading groups and books domains are described in detail. Then system is overviewed. In the section 3.3 system architecture and components presented and in the last section implementation details are mentioned.

3.1 Motivation

Recommender Systems research area is a hot topic in artificial intelligence area. Also recommender systems are highly needed in commercial or non commercial domains. Group Recommender Systems research is a less studied topic among general recommendation research topics. Unlike individual recommenders area, there are quite a few studies are present in this research area.

There are examples of group recommenders in movie, music, restaurant, tourism and touristic attractions etc. domains. However there is no group recommender which intends to help reading groups to pick up relevant books to read during reading season. Challenges and promises of this new domain still stay untouched. We believe that we can gain a lot of knowledge and experience which we can share rest of researchers who are working in recommender systems area by working on this domain. We also believe that we can help real life reading groups a lot with our work BoRGo.

After recommendation process is another new research topic in recommender systems research area. Users need to learn justifications behind recommendation lists and share their thoughts with each other. This kind of need is more severe in group recommendation. Yet there is not enough study in this area. Recommenders systems should serve their users with mediation domains which users could use to reach a consensus over the recommendation list.

All these challenges impelled us to study over a web based group recommender which recommends books to reading groups and supply a mediation domain for after recommendation process.

3.2 Domain Information

Nowadays there are several reading groups which people join for reading and discussing books with other people. In general, reading groups spend a great deal of effort in choosing the books which they will read during the year. Most reading groups have a special session to choose their books before the start of the year.

[56] presents surveys about reading groups' book selection methods. In most of the groups, group members choose what to read altogether in some groups each member chooses one of the books from a list. Some groups use a rotation method. In the simplest version of this method, the host of that session chooses what to read and discuss in the next session. Some groups use a voting method. In this method every member brings a list of books to read and then every member gives as many votes as s/he wants to any book. At the end the top n books, based on number of votes, are read. Most groups arrange a committee to choose the books first. Some other methods can also be found.

In the survey [56] it is seen that nearly the half of the groups choose the entire booklist for the year at once. Most of the reading groups discuss nine books in a year. They mostly do not gather during the summer holidays.

[57–59] are the examples of reading groups in Turkey. Thyke [56] has twenty seven reading groups in two different countries in eight different cities.

Even METU has a reading group in Ankara campus. Members of the reading group in the university first choose books to read and after reading each book, they hold monthly discussion sessions. Choosing the books which most members would find interesting can be difficult for them. It is particularly hard to arrange meetings during the summer holidays, but they need to do this in order to select the books which they will read in the following year.

Supplying a web based recommender which recommends nine books for each month of reading season and has a mediation interface for after recommendation process, to these kind of groups is an innovation.

Books are suitable items for content based recommenders. Because they have sufficient content information which can be used in a content based recommendation algorithm.

3.3 System Overview

BoRGo is a content based group recommender which recommends books for reading groups as well as individual readers. (Figure 3.1) shows flow chart of BoRGo. Novel issues of BoRGo are second filtering method which we used in BoRGo, reading group domain and mediation feature. BoRGo is a web based application. Users can sign in BoRGo with their userids and passwords. BoRGo is capable of ;

- Creating user profiles
- Creating book profiles
- Updating user profiles
- Producing recommendation for individuals
- Creating groups
- Producing recommendation for groups
- Mediation for after recommendation process

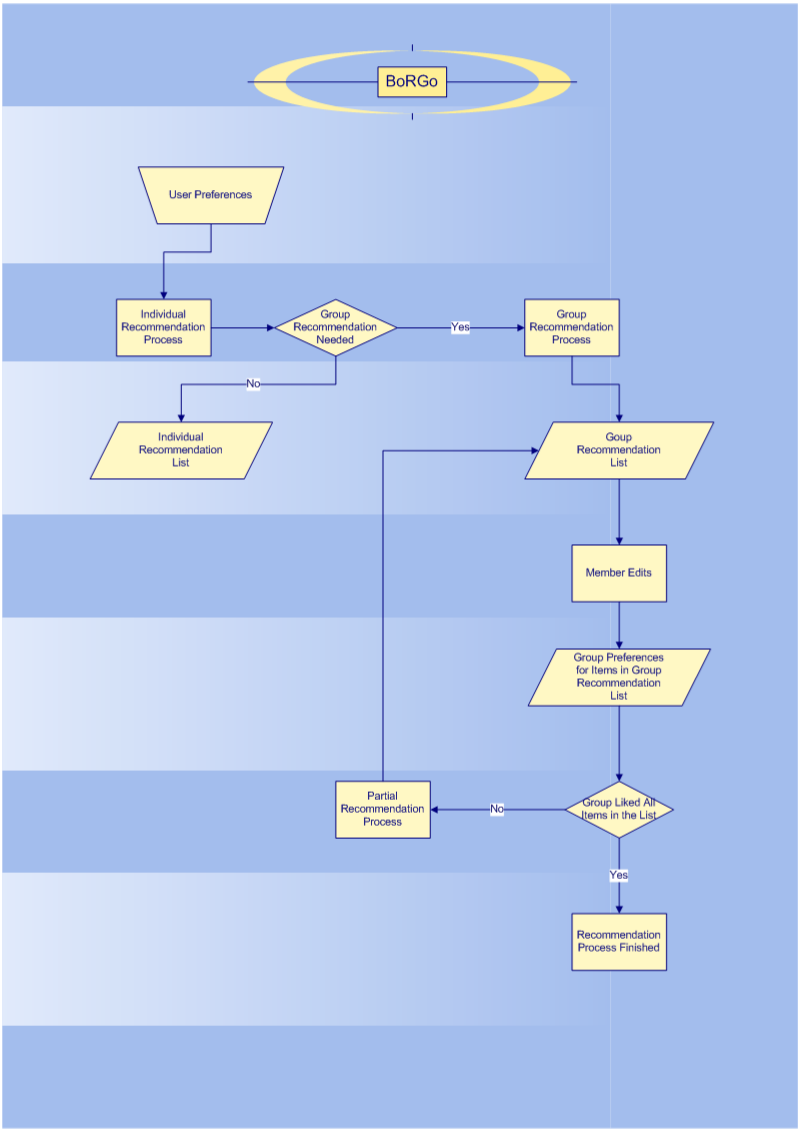


Figure 3.1: Flow chart of BoRGo

BoRGo is a web based recommender system. Users can sign in to the system with their user-ids and passwords.(Figure 3.2) shows web page organization of BoRGo. We used same theme [75] for all of the web pages in BoRGo.

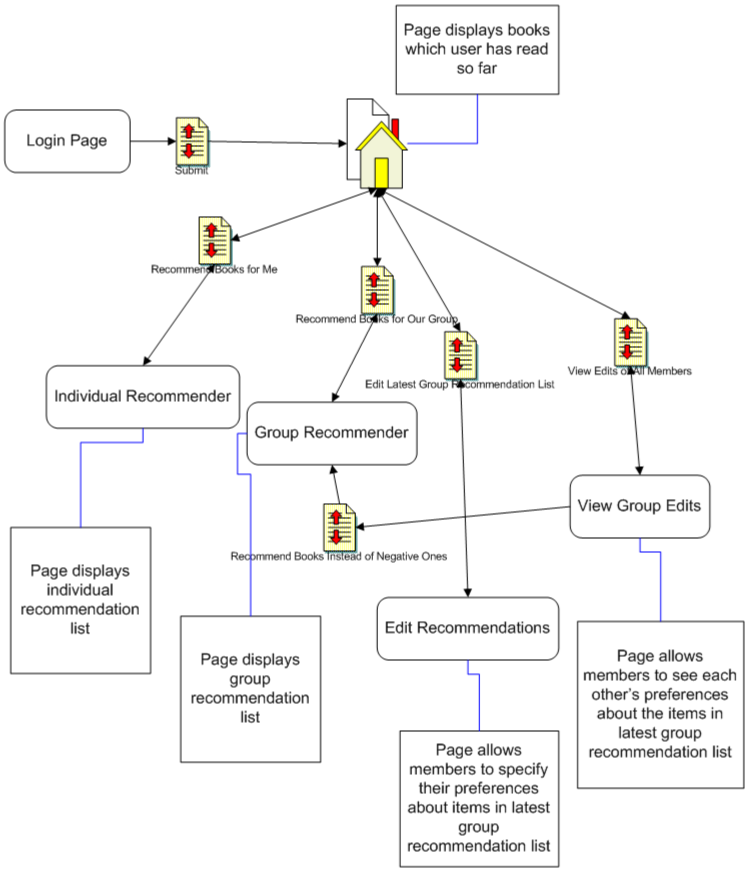
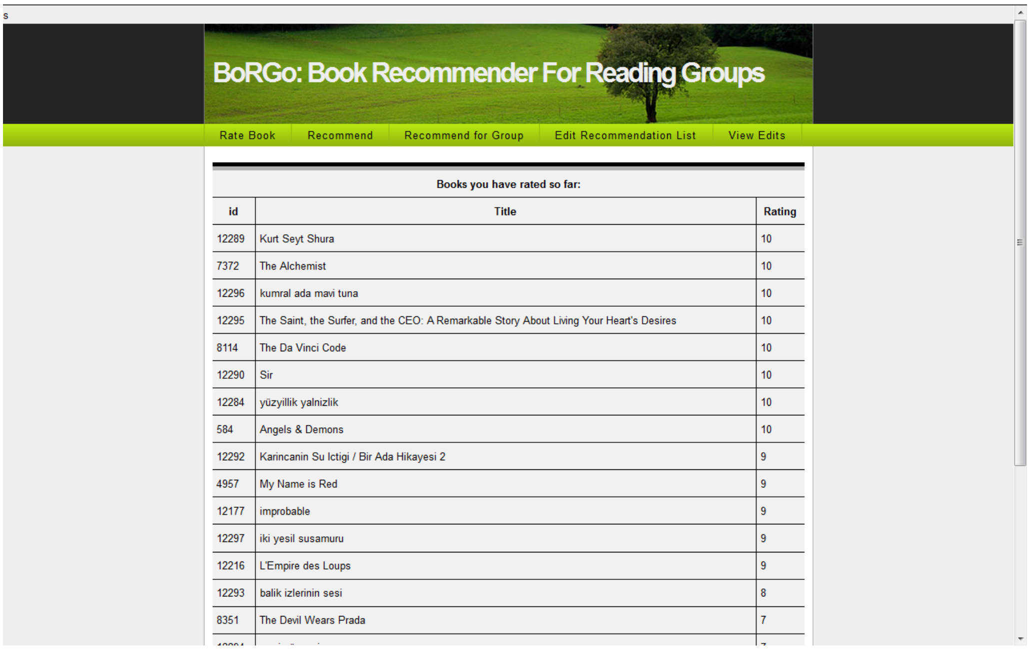


Figure 3.2: Web page organization of BoRGo

After signing in a user first sees the books he rated so far in the screen. Then he can rate new books or ask for recommendation either for himself or group which he belongs to.



id	Title	Rating
12289	Kurt Seyt Shura	10
7372	The Alchemist	10
12296	kumral ada mavi tuna	10
12295	The Saint, the Surfer, and the CEO: A Remarkable Story About Living Your Heart's Desires	10
8114	The Da Vinci Code	10
12290	Sir	10
12284	yüzyillik yalnizlik	10
584	Angels & Demons	10
12292	Karincanin Su Ictigi / Bir Ada Hikayesi 2	9
4957	My Name is Red	9
12177	improbable	9
12297	iki yesil susamuru	9
12216	L'Empire des Loups	9
12293	balik izlerinin sesi	8
8351	The Devil Wears Prada	7

Figure 3.3: Rating history of the user

System architecture of BoRGo will be explained in details in the following parts of this chapter.

3.4 System Architecture And Implementation Details

BoRGo consist of three main components which are;

- the individual recommender,
- the aggregator,
- and the mediator.

Work principals and implementation details of these components will be explained in following subsections.

3.4.1 Individual Recommender

BoRGo can generate recommendations for individuals as well as groups. Indeed individual recommendation process is a main part of group recommendation process also. Therefore explaining this component is very important in terms of understandability of the group recommendation process.

BoRGo is a content based recommender which uses content information of the books which have been rated by the user. Like all other content based recommenders BoRGo has user and book profiles.

Individual recommender finds the books which can possibly satisfy the user mostly, using the user and item profiles. Satisfaction function which shows the satisfaction which user u gets from a book b for individual recommendation is;

$$s : UxB \rightarrow R \tag{3.1}$$

where U is set of users, B is set of books and R $\in$ [0,10].

Recommendation process of individual recommender can be formulated as ;

$$\forall u \in U, b \in B, b'_{\mathrm{u}} = argmaxs(u, b) \tag{3.2}$$

where U is set of users, B is set of books.

3.4.1.1 User Profile

A user profile is like a database which stores preferences or needs of a user [60]. There are two main approaches which can be used for modeling a user [61];

- explicit user modeling
- implicit user modeling

In explicit approaches information about the user is collected directly from the user using keywords, surveys,reviews and feedbacks. Users have full control over the information amount which they gave to the system in these approaches. However explicit user profiling is a time consuming and sometimes boring activity for users.

In implicit approaches information about the user is collected indirectly from communication logs, documents, messages, web logs etc. Unlike explicit approaches implicit approaches minimizes the time and effort which are spent for profiling activities but causes some privacy issues [61]. Because the users have no control over the profiling activity they often feel disturbed.

BoRGo uses explicit user modeling. User profile which constructed by the ratings which the user gave to the books which he already read.

In content based recommendation user profiles can be represented as a vector of feature-rating pairs $f_i{:}w_i$ where f_i is a feature from the domain and w_i is the weight of this feature for the user. BoRGo also uses vectors of feature-rating pairs in user profiles.

User profiles are stored in composition of three different tables in database of BoRGo. ”Users” table stores main information, location information and demographic information about the user.(Figure 3.4) shows ”Users” table.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	**UserID**	**INT**	**11**	☐	☐	☐	**0**
2	Location	VARCHAR	250		☑		NULL
3	Age	INT	11	☐	☑	☐	NULL
4	Password	VARCHAR	10		☑		NULL
5	City	VARCHAR	100		☑		NULL
6	State	VARCHAR	100		☑		NULL
7	Country	VARCHAR	100		☑		NULL
8	EMail	VARCHAR	45		☑		NULL

Figure 3.4: Users table

In ”Users” table ”UserID” column stores unique user ids given to the users. ”Password” column stores passwords of users. ”Age” column stores age information of users. ”Email” column stores email addresses of the users. ”Location” column stores both country, state and city information about users, used in transferring user data from user dataset [62] to database and does not needed anymore. ”Country” column stores country information of the location of the users. ”State” column stores state information of the location of the users. ”City” column stores city information of the location of the users.

”BookRatings” table mainly stores the ratings which user gave to the books which they already read. (Figure 3.5) shows ”BookRatings” table.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	UserID	INT	11	☐	☐	☐	0
2	ISBN	VARCHAR	13		☐		''
3	BookRating	INT	11	☐	☐	☐	0
4	BookID	INT	11	☐	☑	☐	NULL
5	Include	BIT	1		☑		No default

Figure 3.5: BookRatings table

In ”BookRatings” table ”UserID” column is a reference to the ”Users” table. ”BookID” column is a reference to the ”Books” table which is mentioned in following sections. These two columns together form a unique constraint for the ”BookRatings” table. ”ISBN” column stores isbn value which is universally unique for each book , used combining user dataset [62] and book dataset [63] and does not needed anymore. ”BookRating” column stores the rating between 0 and 10 which the user gave to the book. ”Include” column stores bit and used in evaluation of the system.

”UserFeatures” table stores profile information which is extracted from ratings user supplied. (Figure 3.6) shows ”UserFeatures” table.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	UserID	INT	11	☐	☐	☐	No default
2	FeatureID	INT	11	☐	☐	☐	No default
3	PositiveWeight	FLOAT		☐	☑	☐	NULL
4	NegativeWeight	FLOAT		☐	☑	☐	NULL
5	TotalWeight	FLOAT		☐	☑	☐	NULL
6	PositiveOccurance	FLOAT		☐	☑	☐	NULL
7	NegativeOccurance	FLOAT		☐	☑	☐	NULL

Figure 3.6: UserFeatures table

In ”UserFeatures” table ”UserID” column is a reference to the ”Users” table. ”FeatureID” column is a reference to the ”Features” table which is mentioned in following sections. These two columns together form a unique constraint for the ”UserFeatures” table. ”PositiveOccurance” column stores number of books which are rated positively by the user and have the feature indicated by the ”FeatureID” column. ”NegativeOccurance” column stores number of books which are rated negatively by the user and have the feature indicated by the ”FeatureID” column. ”PositiveWeight” column stores the value which is calculated by multiplying ”PositiveOccurrance” value of a feature by ”FeatureScore” of the feature indicated by the ”FeatureID” column. ”FeatureScore” is mentioned detailly in following section. ”NegativeWeight” column stores the value which is calculated by multiplying ”NegativeOccurrance” value of a feature by ”FeatureScore” of the feature indicated by the ”FeatureID” column. ”TotalWeight” column stores the value calculated by adding ”NegativeWeight” and ”PositiveWeight” values and used in BoRGo with the first filtering method.

Ratings higher than 6 are considered as positive ratings and ratings less than 7 are considered as negative ratings in BoRGo. 6 and 5 could have been considered as positive ratings also, but in order to find and recommend the most relevant items it is decided to draw the line which separates positive and negative ratings a bit higher and thus 7 is chosen as the touchstone.

When a user rates a new book, first features of the book are found. Then ”UserFeatures” table is checked for each feature of the book. If there is already an entry for that user, feature pair in the table and if the rating given by the user is positive, ”PositiveOccurance”, ”PositiveWeight” and ”TotalWeight” colums are updated. If the rating is negative then ”NegativeOccurance”, ”NegativeWeight” and ”TotalWeight” colums are updated. If there in no entry for given user, feature pair in ”UserFeatures” table, then a row for that user, feature pair with caluclated ”PositiveOccurance”, ”PositiveWeight”,”NegativeOccurance”, ”NegativeWeight” and ”TotalWeight” values is inserted to the table.

3.4.1.2 Book Profile

In BoRGo each book can have several features which belong to different dimensions. BoRGo uses content knowledge of the books collected from book dataset [63]. This content knowledge is represented by features and dimensions of them.

For example ”Author” is a dimension in BoRGo and ”Adam Fawer” is a feature value of ”Author” dimension. ”Language” is another dimension and ”Turkish” is a feature of ”Language” dimension.

Book profiles are stored in composition of four different tables in database of BoRGo.

"Books" table stores main information about the book.(Figure 3.7) shows "Books" table.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	Name	VARCHAR	200		☐		No default
2	Title	VARCHAR	200		☑		NULL
3	id	INT	11	☐	☐	☐	AUTO_INCREMENT

Figure 3.7: Books table

In "Books" table "ID" column stores the unique id which is given by the system to the book. "Name" column stores the name of the book in the form which it appeared in book dataset [63] and does not used any more. "Title" column stores the title of the book.

"Dimensions" table stores dimension information. (Figure 3.8) shows "Dimensions" table.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	Name	VARCHAR	150		☐		No default
2	id	INT	11	☐	☐	☐	AUTO_INCREMENT

Figure 3.8: Dimensions table

In "Dimensions" table, "ID" column stores the unique id which is given by the system to the dimension. "Name" column stores name of the dimension.

"Features" table stores the features. Each feature is added only once to this table. (Figure 3.9) shows "Features" table.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	dimensionID	INT	11	☐	☐	☐	No default
2	Value	VARCHAR	300		☐		No default
3	FeatureScore	FLOAT		☐	☑	☐	NULL
4	**id**	**INT**	**11**	☐	☐	☐	**AUTO_INCREMENT**

Figure 3.9: Features table

In ”Features” table ”ID” column stores the unique id which is given by the system to the feature. ”DimensionID” column is a reference to the dimension which feature belongs to. ”Value” column stores value of the feature. Each feature has a feature score which identifies the importance of that feature. This feature score is stored in ”FeatureScore” column of ”Features” table in the database.

Some features are more important than others to distinguish a book. We propose that, a feature f_i will be more discriminative if fewer books in the database have it. The more features a dimension of f_i has, the more discriminative the feature f_i will be. To model these rules, we used a TF-IDF measure [64]. We calculate the feature score FS of each feature f_i with the formula (Equation 3.3)

$$FS(f_i) = log(B/Bf_i) \times log(D_j) \tag{3.3}$$

where B is the total book count, Bf_i is the count of the books which have feature f_i and D_j is the count of the features which belongs to the dimension D_j which feature f_i belongs to.

Calculation of feature scores is done initially when the book dataset [63] transferred to the database. When a new book is added to the BoRGo all of the feature scores are recalculated. Then ”UserFeature”s table is updated because of the changes in the feature scores.

Finally ”BookFeatures” table stores relations between ”Books” and ”Features”. A feature or a book can be added to this table several times because more than one book can have same feature and a book can have more than one feature. (Figure 3.10)shows ”BookFeatures” table.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	BookID	INT	11				No default
2	FeatureID	INT	11				No default
3	**id**	**INT**	**11**				**AUTO_INCREMENT**

Figure 3.10: BookFeatures table

In ”BookFeatures” table ”ID” column stores the unique id which is given by the system to the ”BookFeature” entity. ”BookID” column is a reference to the ”Book” table. ”FeatureID” column is a reference to the ”Feature” table.

3.4.1.3 Individual Recommendation Process And Filtering Method

BoRGo is capable of producing individual recommendation lists. These individual lists are also base of the group recommendation process. For individual recommendation we used two different filtering methods in BoRGo.

BoRGo differs from other content based recommenders with the second and actual filtering method used in it.

First Filtering Method Used in BoRGo As it is mentioned in Section 3.3.1.1 each user feature pair of user profiles has ”PositiveWeight” and ”NegativeWeight” values. Content based recommenders like OpenMore [65–67] calculate the ”TotalWeight” of a feature by adding the ”PositiveWeight” and the ”NegativeWeight” values of the feature and use this ”TotalWeight” value while calculating recommendation scores. We also used this recommendation score calculation method in the first filtering method of BoRGo.

In the first filtering method, we considered the books which had at least one common feature with the books that were rated (either positively or negatively) by the user, as candidate books.

The final recommendation score of a book was calculated as the sum of the ”TotalWeight” of the common features between the candidate book and the user profile. We calculated the ”TotalWeight” of a feature by adding the ”PositiveWeight” to the ”NegativeWeight” of that feature. (Figure 3.11) shows the pseudo code of this process.

```
Calculate_Recommendation_Score (CandidateBookSet CB, UserProfile up)
{
        for each cb ∈ CB
        {
                CommonFeatures CF = Find_Common_Features( cb , up );
                Number recommendationScore = 0;

                for each cf ∈ CF
                {
                        Number totalWeight = Positive_Weight_Of( cf ) + Negative_Weight_Of( cf );
                        recommendationScore = recommendationScore + totalWeight;

                }
                Recommendation_Score_Of( cb ) = recommendationScore;
        }
}
```

Figure 3.11: Calculate Recommendation Score algorithm

After their recommendation scores are calculated, candidate books are sorted according to their scores. Finally, a recommendation list is presented to the user.

Second Filtering Method Used in BoRGo In second filtering method we changed both the way of finding candidate book set -our filtering method- and calculation of recommendation scores for these candidate books. In this filtering method only the books which have more potential of being liked by the user are chosen to be candidates. We also decided to use ”AbsoluteTotalWeight” instead of ”TotalWeight” in this version. When using ”TotalWeight”, negative ratings are used in a positive manner in the calculation of the weight of a feature. In contrast, we can use negative ratings in a negative manner by using ”AbsoluteTotalWeight”, because of the fact that we calculate the ”AbsoluteTotalWeight” of a feature by subtracting the ”NegativeWeight” of that feature from the ”PositiveWeight”.

In second filtering method we changed our filtering method and considered the books which had at least one common feature with the books that were rated positively by the user to calculate a recommendation score for a candidate book.

After finding candidate books set BoRGo first calculates the ”AbsoluteTotalWeight” of the features which are common to both the user profile and the candidate book profile. The ”AbsoluteTotalWeight” of a feature can be defined as the difference between the ”PositiveWeight” and the ”NegativeWeight” of that feature for that specific user.

After calculating the ”AbsoluteTotalWeights” of the common features, BoRGo takes the sum of the ”AbsoluteTotalWeights” of these common features to calculate a final recommendation score for that book. (Figure 3.12) shows the pseudo code of this process.

```
Calculate_Recommendation_Score_2 (CandidateBookSet CB, UserProfile up)
{
        for each cb ϵ CB
        {
                CommonFeatures CF = Find_Common_Features( cb , up );
                Number recommendationScore = 0;

                for each cf ϵ CF
                {
                        Number absoluteTotalWeight = Positive_Weight_Of( cf ) – Negative_Weight_Of( cf );
                        recommendationScore = recommendationScore + absoluteTotalWeight;

                }
                Recommendation_Score_Of( cb ) = recommendationScore;
        }
}
```

Figure 3.12: Calculate Recommendation Score 2 algorithm

After their recommendation scores are calculated, the candidate books are sorted according to their scores. Finally, a recommendation list is presented to the user.

3.4.2 Group Recommender

A user can ask for recommendation for the group he belongs to when he sign in to the system. Group profile is stored in two different tables in database. ”Groups” table stores the main information about the group. (Figure 3.13)shows ”Groups” table.

In ”Groups” table, ”ID” column stores the unique id which is given by the system to the group. ”Name” column stores name of the group. ”Description” column stores description of the group.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	**ID**	**INT**	**11**	☐	☐	☐	**AUTO_INCREMENT**
2	Name	VARCHAR	50		☑		NULL
3	Description	VARCHAR	250		☑		NULL

Figure 3.13: Groups table

”GroupUsers” table stores relation between group and its members. ”GroupID” column is a reference to ”Groups” table and ”UserID” column is a reference to ”Users” table. ”IsModerator” column stores the moderator of the group. (Figure 3.14)shows ”GroupUsers” table.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	GroupID	INT	11	☐	☐	☐	No default
2	UserID	INT	11	☐	☐	☐	No default
3	IsModerator	BINARY	1		☑		NULL

Figure 3.14: GroupUsers table

Group recommendation process needs an aggregation algorithm and satisfaction function. As it is mentioned earlier there are two different aggregation algorithms and a few different satisfaction functions which are used in group recommenders [30–37]. Recommendation algorithm and satisfaction function of BoRGo is explained in the next subsection.

3.4.2.1 Aggregator

After a group is formed and one of the members of the groups asks for the group recommendation from BoRGo, BoRGo first generates individual recommendation lists for each member of the group like it is mentioned in Section 3.3.1.3. Then it aggregates individual recommendation lists into a final recommendation list for the group. As we mentioned before this method is called joint recommendation list method. With this method it is easy to explain the final list and the order of the books in the list to the users. This is the main factor for us to choose to use this method in aggregation process.

The most important drawback of this method is that it slows down the recommender when the group is too big in size. But in nature reading groups generally consist of 5-10 people and this size is small enough to allow usage of joint recommendation list method.

There are several satisfaction functions that can be applied through the aggregation process that we mentioned in Section 2.3.2.2. We chose ”average user satisfaction” in order to satisfy the majority of the group members to a certain degree. [52] shows that using ”average user satisfaction” can upset some of the members of the group who do not rate books as much as others or who have marginal tastes. However in BoRGo, we prefer to satisfy the majority of the group happy instead of careing for marginal or non-contributing members.

When the user asks for recommendation for a group, BoRGo generates individual recommendation lists as mentioned in Section 3.3.1.3. Then group scores for each distinct book in individual lists are calculated. The group score for book b_i which is calculated for each book selected from individual lists, is as shown in the following formula ;

$$GS(b_i) = \sum_{j=0}^{n} IS(b_{ij})/MC \tag{3.4}$$

In above formula, GS(b_i) is calculated group score for book b_i, IS(b_{ij}) is the individual recommendation score of book b_i for member j. If b_i is not in one of the group members' recommendation lists then the individual score of that book for that member is zero. MC is the member count of the group.

After group scores are calculated for each book, books are sorted according to their group scores and the first nine books are presented as the group recommendation list to the users. A final recommendation list of nine books is appropriate for the nature of our test groups and also real reading groups because of the fact that groups often do not meet during summer holidays and thus they discuss only nine books per year.

Example (Table 3.1) , (Table 3.2) and (Table 3.3) show sample individual recommendation lists for members of a group.

Table 3.1: Individual recommendation list for the Member1

Book Title	Recommendation Score
Dragon Prince	10
First King of Shannara	9
Sisters of Isis	8
The Unexpected	7
A Talent for War	7
Dead Souls	6
The Gladiator	4
To Play the Fool	4
Treasure Island	3

Table 3.2: Individual recommendation list for the Member2

Book Title	Recommendation Score
Port of Saints	10
Dragon Prince	8
Dead Souls	7
Impressions	7
Sisters of Isis	6
Treasure Island	5
Chronicles of Pern: First Fall	4
Ashes of Eden	3
Tribulations of a Chinaman in China	1

Table 3.3: Individual recommendation list for the Member3

Book Title	Recommendation Score
Impressions	9
Meet the Austins	7
A Talent for War	7
Sisters of Isis	6
First King of Shannara	6
Dragon Prince	5
Chronicles of Pern: First Fall	4
To Play the Fool	3
Ashes of Eden	3

Group recommendation scores of each book in individual recommendation lists are calculated. For example for the book ”Dragon Prince” group recommendation score is calculated as (10 + 8 + 5) / 3 = 7 (Table 3.4) shows group recommendation scores of books in individual lists.

Table 3.4: Group recommendation scores which are calculated for all of the books in individual recommendation lists of group members. RS indicates Recommendation Score

Book Title	RS for Member1	RS for Member2	RS for Member3	RS for Group
Dragon Prince	10	8	5	7
First King of Shannara	9	6	-	5
Sisters of Isis	8	6	6	6
The Unexpected	7	-	-	2
A Talent for War	7	7	-	4
Dead Souls	6	7	-	4
The Gladiator	4	-	-	1
To Play the Fool	4	3	-	2
Treasure Island	3	5	-	2
Port of Saints	-	10	-	3
Impressions	-	7	9	5
Chronicles of Pern: First Fall	-	4	4	2
Ashes of Eden	-	3	3	2
Tribulations of a Chinaman in China	-	1	-	0
Meet the Austins	-	-	7	2

After calculation of group recommendation scores books are ordered by their group recommendation scores and first nine books are presented to the group by the BoRGo. (Table 3.5) shows final group recommendation list for this example.

Table 3.5: Group recommendation list for the example reading group

Book Title	Group Recommendation Score
Dragon Prince	7
Sisters of Isis	6
First King of Shannara	5
Impressions	5
A Talent for War	4
Dead Souls	4
Port of Saints	3
The Unexpected	2
To Play the Fool	2

3.4.2.2 Mediator

Recommendation process does not come to an end after recommendation list is presented to the user(s). Especially in group recommendation, members of the group need to reach a consensus over the list. This is the same for reading groups domain. In natural way of deciding which books to read -without a recommender- reading groups use some different techniques.

BoRGo -as a recommender system- helps reading groups to find most relevant items for the group. In addition to find relevant books BoRGo also supply a media to the users which they can share their thoughts of recommended books with the other group members. Also moderator of the group can ask for new recommendation only instead of the books which are not accepted by the group. These two novel features of BoRGo works as a mediator for after recommendation process.

Some other group recommenders like PolyLens [32], Pocket Restaurant Finder [33] and [33] addresses after recommendation process but none of them supply a media where members of the group can share their opinions about recommended items and ask for partial recommendation.

BoRGo presents two different user interfaces for after recommendation process. First of them is the "Edit Group Recommendation" screen that each member can see the books in the latest group recommendation list. Members can give feedback about each book in the list from this screen. (Figure 3.15) shows "Edit Group Recommendation".

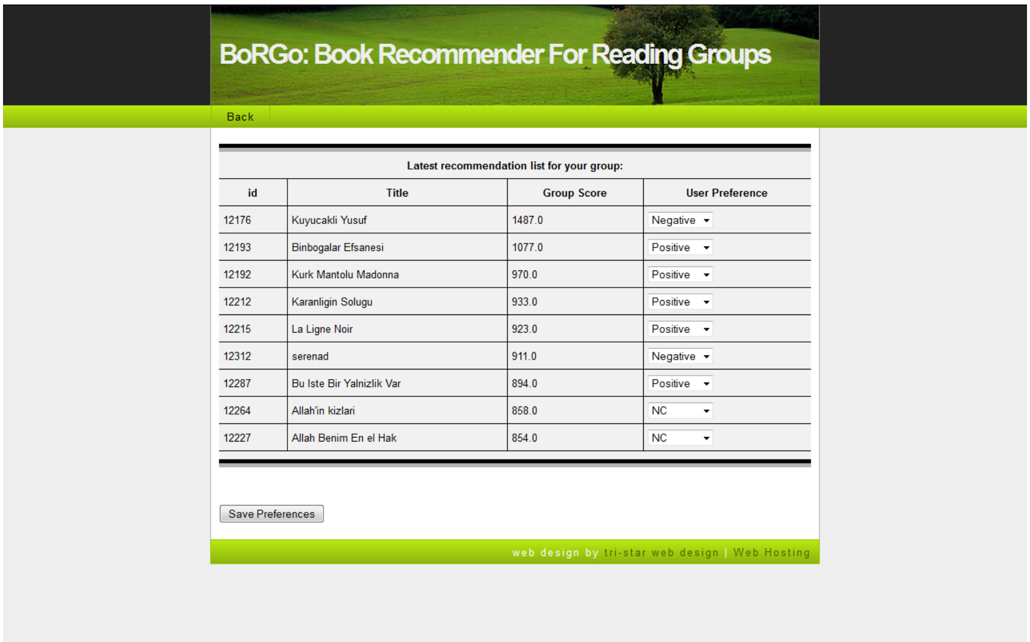


Figure 3.15: Edit Group Recommendation screen

After the group recommendation list is produced. Each member of the group can see recommended books from "Edit Group Recommendation" screen. In the screen users can see id number , title and group score of the recommended books and specify his preferences about the book.

Default value of user preferences for the books is "DC" which means "Don't Care". If user remains his preference for the book as "DC" , it means that user does not care to read or not to read that book. Instead of "DC" user can specify his preference about the book as "Positive" or "Negative" by choosing from the "User Preference" combo for the book. When a user gives a negative feedback about a book via this screen, it changes that user's profile information.

After selecting a preference value for each book in the recommendation list, user must save his preferences. User can see or change his preferences from same screen afterwards.

Second interface of the mediator is the screen which shows the preferences of all of the group members together. With this screen each member of the group can see each other's thoughts for the books in the group recommendation list. (Figure 3.16) shows "Member Preferences" screen.

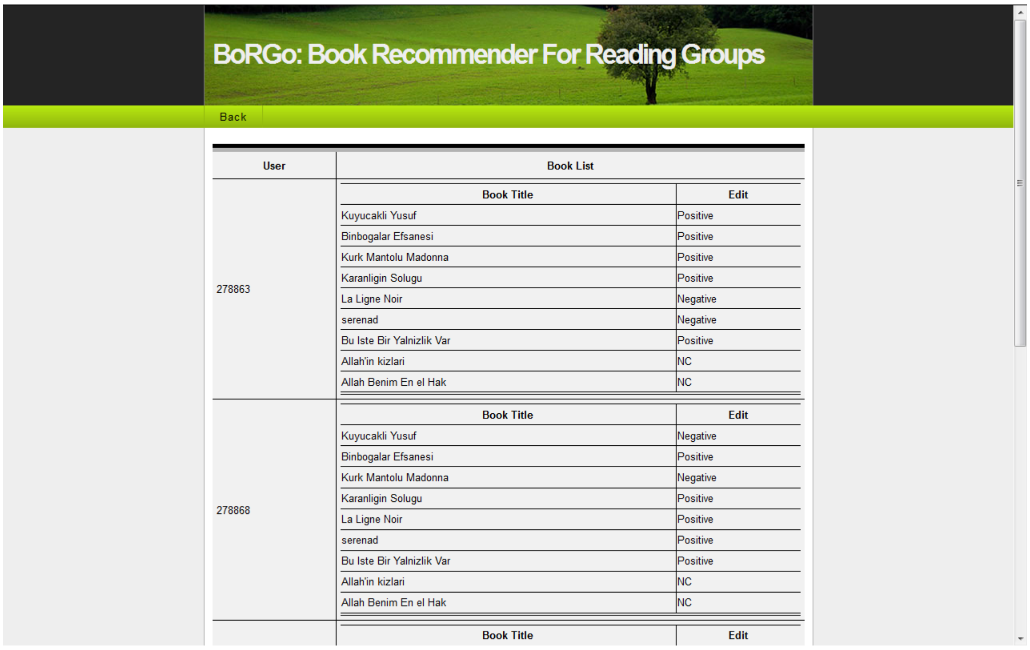


Figure 3.16: "Member Preferences" screen

"Member Preferences" screen also helps moderator to see which members of the group did not indicate their preferences about the group recommendation list. Moderator can send notifications to the members which did not give any thoughts about any book in the list.

After each member shares their thoughts about the books in the list, if it is necessary moderator of the group can ask new recommendations only for the books in the recommendation list which at least half of the group members did not voted positively. This type of recommendation request does not change the whole list instead BoRGo finds the books in the list which at least half of the group members voted negatively and recommends new books instead of them. During this process "DC"s are considered as positive feedbacks for the books.

New group recommendation list is presented to the members via "Edit Group Recommendation" screen with newly recommended books. Members can give feedback to the new books in the list via this screen. After everybody finished editing new books, if it is needed moderator can ask for new recommendation again until all of the books in the list are indicated positively at least by half of the group members.

(Figure 3.17) shows pseudo code for finding which books will be removed from group recommendation list.

```
Find_Negatively_Rated_Books (RecommendedBooks RB, UserEdits UE, Number memberCount)
{
        Array negativelyRatedBooks;

        FOR EACH rb ϵ RB
        {
                Number negativeEdits = 0;
                FOR EACH ue ϵ UE
                {
                        IF Get_Book_ID( rb ) = Get_Book_ID( ue ) THEN
                                IF Is_Negative( ue ) THEN
                                        negativeEdits = negativeEdits + 1;
                }

                IF negativeEdits > memberCount / 2 THEN
                        Add( negativelyRatedBooks , rb );

        }

        return negativelyRatedBooks;
}
```

Figure 3.17: Finding negatively rated books

CHAPTER 4

EVALUATION OF THE SYSTEM

This chapter presents evaluation details of the BoRGo. Firstly user and book datasets are reviewed. Then evaluation metrics are explained. Lastly evaluation details and results experiments of BoRGo are presented.

4.1 Data Set

We used combination of two different datasets -one for books and one for user ratings- in evaluation of BoRGo. User dataset [62] contains UserID, Age and Location information about 278859 users and 1149781 ratings scaled from 0 to 10 which these users gave for books.

Book dataset [63] contains various information in 82 different dimensions about books from author to image caption of the book. (Table 4.1) shows list of dimensions which book dataset contains.

Table 4.1: Alphabetically grouped List of dimensions in book dataset

Dimension Name
actor, align, alternateName,asin,author
bgcolour, bookDesign
callNumber, caption, characters, classification, country, cover, coverArtist, coverDesigner, coverPhotographer
death, dedicatedTo
editedBy, edition, editor, endings, englishPubDate, englishReleaseDate, era
filmVersion ,followedBy
galacticYear, genre
id, illustrator, imageCaption, imageSize, imagewidth, infoboxwidth, introductionBy, isbn
jacketDesign, jacketPhotograph
language, lastBook, laterTitle, length, literaryAward, locations
medaType, mediaType, motifs
narrator, notes, numberInSeries, numberOneFan
oclc
pages, photographer, picture, portraitArtist, precededBy, prefaceBy , price, pubDate, publicationDate, publicationType, publishedIn, publisher
releaseDate, revisedName, runningTime
sales, series, size, starring, structure, subject, subtitle, succeededBy
theme, titleOrig, translator
website, width, words

However each book does not have same type of information. (Figure 4.1) shows two different books in the dataset. In this example the book titled ”2061 Odyssey Three” has information about illustrator and ISBN but the book titled ”2001 A Space Odyssey” does not have information for those two dimensions. Instead it has information about image caption.

```
2061:_Odyssey_Three:
  name: "2061: Odyssey Three"
  isbn: ISBN 0-345-35173-8
  language: "[[English_language]]"
  releaseDate: "1987"
  precededBy: "[[2010:_Odyssey_Two]]"
  author: "[[Arthur_C._Clarke]]"
  country: "[[United_States]]"
  series: "[[The_Space_Odyssey_series]]"
  publisher: "[[Del_Rey_Books]]"
  genre: "[[Science_fiction_novel]]"
  followedBy: "[[3001:_The_Final_Odyssey]]
  mediaType: Print
  pages: 256 pages
  illustrator: "[[Michael_Whelan]]"
```

```
2001:_A_Space_Odyssey_(novel):
  name: "2001: A Space Odyssey"
  isbn: NA
  language: "[[English_language]]"
  releaseDate: "1968"
  imageCaption: Dust-jacket from the first edition
  author: "[[Arthur_C._Clarke]]"
  country: "[[United_States]]"
  series: "[[The_Space_Odyssey_series]]"
  publisher: "[[New_American_Library]]"
  genre: "[[Science_fiction]]"
  followedBy: "[[2010:_Odyssey_Two]]"
  mediaType: Print
  pages: 221 pp (first edition, hardback)
```


Figure 4.1: Two examples from the book dataset.

Books are uniquely identified by ISBN numbers in both datasets. ISBN is defined as ”The International Standard Book Number (ISBN) is a 10-digit number that uniquely identifies books and book-like products published internationally.” in [71]. We combined two datasets over ISBN.

When we try to match up books from book dataset to the books from user dataset over ISBN, we saw that some of the ISBNs does not match each other or some books in book dataset already does not have a ISBN.

Although book dataset contains 12322 books and user dataset contains 1149781 ratings originally, combination of these two datasets contains only 17270 users whom rated at least one book and 54017 ratings for 3230 different books because of lack of ISBN matching. (Table 4.2) shows distribution of the ratings to the users.

Table 4.2: Rating counts of users

User Count	Rating Count
16.373	between 1 and 10
505	between 11 and 20
185	between 21 and 30
84	between 31 and 40
41	between 41 and 50
82	at least 50

4.2 Evaluation Metrics

BoRGo has a database which includes books, users and the ratings that are assigned to books by the users. It is a sparse database in terms of ratings. In all of the following experiments except user studies we calculated the rating of book b for the group as in the formula (Equation 4.1);

$$R_{\text{group}} = (R_1 + R_2 + ...R_{\text{n}})/n \tag{4.1}$$

In this formula R_{n} is the rating given for that book by the member n. If a member did not rate that book, it would not affect the group's rating for that book. In other terms we only considered existing ratings to calculate the rating of the book b for the entire group.

We did 3-fold cross validation in order to evaluate BoRGo in each experiment. We randomly took 1/3 of the books as our test group in each validation, and tried to see which of those books would be recommended to the group.

We used categorization of the items which is shown in the (Table 4.3)in calculation of the evaluation metrics.

Table 4.3: Categorization of items

	Actual Positive	Actual Negative
Predicted As Positive	TP	FP
Predicted As Negative	FN	TN

We considered the books whose group's ratings were positive (actual positive) and recommended to the group (predicted as positive) as true positives (TP), the books whose group's ratings were positive (actual positive) but were not recommended to the group (predicted as negative) as false negatives (FN), the books whose group's rating were negative (actual negative) and were not recommended to the group (predicted as negative) as true negatives (TN) and the books whose group's rating are negative (actual negative) and recommended to the users (predicted as positive) as false positives (FP).

Then we calculated the precision, recall, accuracy and F-measure metrics for each experiment by using the formulas (Equation 4.2), (Equation 4.3), (Equation 4.4) and (Equation 4.5).

It can be said that Precision and Recall are the most common metrics used in evaluation recommender systems [68–70].

Precision can be defined as the ratio of number of relevant items selected correctly to the number of items selected and represents the probability of a selected item is really relevant for a user.

$$Precision = TP/(TP + FP) \tag{4.2}$$

Recall can be defined as the ratio of relevant items selected to the total number of relevant items which are available and represents the probability of a relevant item to be selected.

$$Recall = TP/(TP + FN) \tag{4.3}$$

Accuracy can be defined as the ratio of correct recommendations to the total possible recommendations.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{4.4}$$

F-Measure combines Precision and Recall metrics into a single metric by giving both to Precision and Recall equal weights.

$$F - Measure = (2 \times Precision \times Recall)/(Precision + Recall) \tag{4.5}$$

4.3 Experiments With Using First Filtering Method

Initially we do not have any group information in datasets. In order to evaluate BoRGo we needed to form the user groups. We thought that users who live in the same city probably join the same reading community. To make the test groups more realistic, we decided to group users who live in the same cities according to their age intervals. We selected the cities which contained more than fifteen users who rated at least one book. Then we grouped people in each city according to their ages. Finally we tested our system with eighteen different groups.(Table 4.4) shows evaluation results of these experiments. In the table PRBC means count of the books which are rated positively by group members and NRBC means Count of the books which are rated negatively by group members.

Table 4.4: Evaluation results of the experiments with first filtering method we used in BoRGo.

Member Count	PRBC	NRBC	Precision	Recall	Accuracy	Fmeasure
10	11	6	0,83	0,64	0,65	0,72
6	5	2	0,67	0,84	0,59	0,74
10	7	14	0,36	0,44	0,49	0,38
7	6	12	0,44	1,00	0,50	0,60
10	6	8	0,50	1,00	0,57	0,67
8	15	9	0,76	0,60	0,61	0,63
14	7	11	0,39	1,00	0,39	0,56
10	11	14	0,41	0,36	0,48	0,37
5	4	21	0,23	1,00	0,41	0,37
7	4	5	0,38	0,75	0,35	0,50
5	4	4	0,53	1,00	0,58	0,69
5	6	12	0,25	0,50	0,47	0,33
12	10	10	0,50	1,00	0,50	0,67
13	24	17	0,76	0,55	0,63	0,63
6	4	11	0,17	0,50	0,20	0,25
8	11	13	0,43	0,75	0,46	0,52
6	8	6	0,83	0,83	0,72	0,78
8	4	10	0,15	0,50	0,22	0,23

These evaluation results are not satisfying in terms of finding relevant items for the groups. Therefore we made some critical changes in BoRGo which are mentioned in Section 3.3.1.3.

4.4 Experiments With Using Second Filtering Method

We evaluated BoRGo with the second filtering method, with the same groups and it is seen that the results improved remarkably compared to results in first experiments. The later results can be seen in (Table4.5). In the table PRBC means count of the books which are rated positively by group members and NRBC means Count of the books which are rated negatively by group members.

Table 4.5: Evaluation results of the experiments with second filtering method we used in BoRGo.

Member Count	PRBC	NRBC	Precision	Recall	Accuracy	Fmeasure
10	11	6	0,93	1,00	0,94	0,96
6	5	2	1,00	1,00	1,00	1,00
10	7	14	1,00	1,00	1,00	1,00
7	6	12	1,00	1,00	1,00	1,00
10	6	8	1,00	1,00	1,00	1,00
8	15	9	1,00	0,87	0,92	0,92
14	7	11	1,00	1,00	1,00	1,00
10	11	14	1,00	1,00	1,00	1,00
5	4	21	0,75	1,00	0,93	0,84
7	4	5	1,00	1,00	1,00	1,00
5	4	4	1,00	1,00	1,00	1,00
5	6	12	0,89	1,00	0,94	0,93
12	10	10	0,93	1,00	0,96	0,96
13	24	17	0,92	0,71	0,78	0,79
6	4	11	1,00	1,00	1,00	1,00
8	11	13	0,92	1,00	0,96	0,95
6	8	6	1,00	0,92	0,94	0,95
8	4	10	1,00	1,00	1,00	1,00

The second filtering method is better than the first filtering method in terms of finding the relevant items. In the first experiments the average precision value was calculated as 0.55, the average recall value was calculated as 0.74, the average accuracy value was calculated as 0.54 and the average f-measure value was calculated as 0.58. In the second set of experiments, in contrast, the average precision value is calculated as 0.96, the average recall value is calculated as 0.97, the average accuracy value is calculated as 0.97 and the average f-measure value is calculated as 0.96.

4.4.1 Evaluation of BoRGo with Different Sized Groups

In order to see the performance of BoRGo with different sized groups we did experiments. For these experiments we composed ten groups with five different group sizes by randomly selecting members.

The interests of the group members within each group may not overlap because of the random member selection mechanism and ratings are very sparse for these groups. You can see the evaluation results of these experiments in (Table4.6)

Table 4.6: Evaluation results of different sized random groups.

Member Count	Precision	Recall	Accuracy	Fmeasure
10	1	1	1	1
10	1	0,83	0,91	0,91
15	1	1	1	1
15	0,83	1	0,94	0,89
20	1	0,73	0,85	0,84
20	1	0,93	0,96	0,96
25	0,95	0,85	0,91	0,87
25	0,83	0,22	0,6	0,38
30	1	0,38	0,81	0,54
30	1	0,63	0,86	0,77

As the group gets larger, recall and F-measure values decrease remarkably. Accuracy also decreases but not that remarkably. Precision remains almost the same. In this experiment the average precision is calculated as 0,96 , the average recall is calculated as 0,76, the average accuracy is calculated as 0,88 and the average F-measure is calculated as 0,81.

4.4.2 Evaluation of BoRGo with the Groups Whose Ratings Are Dense

We formed five different groups with members selected randomly among the users who rated highly rated books. These groups' sizes vary. The ratings are mostly very dense, and the users' tastes overlap for these group combinations.

The groups which we used in Section 4.4.1 have ratings between 11 and 186. On the other hand, the groups we used in this experiment have ratings between 589 and 1134 in total. The results of this experiment are shown in (Table4.7)

Table 4.7: Evaluation results of groups with dense ratings.

Member Count	Precision	Recall	Accuracy	Fmeasure
4	0,92	0,08	0,58	0,4
4	0,83	0,08	0,59	0,15
5	0,85	0,08	0,75	0,15
10	0,8	0,09	0,78	0,16
14	0,81	0,05	0,69	0,1

In this experiment the average precision value is calculated as 0,84, the average recall value is calculated as 0,08, the average accuracy value is calculated as 0,68 and the average F-measure value is calculated as 0,19. The reason behind these low recall, accuracy and F-measure values is that our recommender only shows nine recommendations to the group . For example member of one of the groups which we used in this experiment rated 510 different books. 242 of these 510 books are rated positively by the members. When we applied 3 fold cross validation for this group, in average we removed 80 positively rated books in each fold but we recommended only 9 books which means number of true positives can be at most 9 and number of false negatives can be at least 71. Therefore metrics which use true positives over false negatives in their formula resulted low values.

4.4.3 User Studies

We asked users for book ratings via facebook [72]. 15 users rated 148 different books in total. We grouped these users into four groups according to their interests. Then we recommended 9 books to each group also via facebook. We asked each user if he/she would like to read each book during the year as a member of that reading group. We assumed that if at least half of the members of a group are willing to read a recommended book, that book is accepted by the whole group. We currently have three of four groups' responses. For group one, we achieved a hundred percent success. All the books we recommended have been accepted by the group. For group two, eight out of nine recommended books were accepted by the group. The third group accepted seven of nine books. So we achieved a 0.89 average success rate.

In previous experiments we studied with user data from users who mostly read in English. But in the user studies reported here, our users are people who frequently read books in Turkish. The majority of the books in our database are written in English. This causes Turkish to be a more discriminative feature and BoRGo takes this account and recommends Turkish books to the users who may prefer an English action book to a Turkish romance book. This explains the difference between success rates of previous experiment and this user study. Importance of a feature is strongly related to the used dataset. If Turkish books were majority in our database, Turkish would not be a more discriminative feature and would not affect the recommendation list this much.

4.5 User Studies For After Recommendation Process

As it is mentioned in Section 3.3.2 we developed a mediator for BoRGo in order to help groups with after recommendation process. With an experiment we wanted to learn that after how many cycles members of a group come a total consensus over the group recommendation list.

12 users participated this study. They are randomly grouped into three different

groups. Users whom participated to this study gave 151 ratings to 122 different books. BoRGo recommended to each group a list of books.

(Table4.8) shows recommendation list of the first group and responses of group members to each item in the list. There are two books which are found to be negative for the group after all of the members identified their opinions about the books in the group recommendation list.

Table 4.8: Recommendation list of first group and member responses to the items.

Book Title	Response of Member1	Response of Member2	Response of Member3	Response of Member4	Response of Group
Kuyucakli Yusuf	Positive	Positive	Positive	Negative	Positive
Allah Benim En el Hak	Negative	Positive	Positive	Negative	Positive
Tanios Kayasi	Positive	Positive	Positive	Positive	Positive
Yollarin Baslangici	Positive	Negative	Positive	Positive	Positive
sehrin aynalari	Negative	Negative	Negative	Negative	Negative
siyah süt	Positive	Negative	Negative	Negative	Negative
Isik Bahceleri	Positive	Positive	Negative	Negative	Positive
dogunun limanlari	Positive	Positive	Negative	Positive	Positive
Karincanin Su Ictigi / Bir Ada Hikayesi 2	Negative	Positive	Positive	Negative	Positive

Then moderator of the group asked for the new recommendation instead of those two books -”sehrin aynalari” and ”siyah süt”. BoRGo recommended the list in the (Table4.9). As it can be seen from the table all of the group members accepted this version of recommendation list.

Table 4.9: Recommendation list of first group after partial recommendation.

Book Title	Response of Member1	Response of Member2	Response of Member3	Response of Member4	Response of Group
Kuyucakli Yusuf	Positive	Positive	Positive	Negative	Positive
Allah Benim En el Hak	Negative	Positive	Positive	Negative	Positive
Tanios Kayasi	Positive	Positive	Positive	Positive	Positive
Yollarin Baslangici	Positive	Negative	Positive	Positive	Positive
Isik Bahceleri	Positive	Positive	Negative	Negative	Positive
dogunun limanlari	Positive	Positive	Negative	Positive	Positive
Karincanin Su Ictigi / Bir Ada Hikayesi 2	Negative	Positive	Positive	Negative	Positive
Binbogalar Efsanesi	Positive	Positive	Positive	Negative	Positive
Allah'in kizlari	Negative	Negative	Positive	Positive	Positive

(Table4.10) shows recommendation list of the second group and responses of group members to each item in the list.This group accepted the first list BoRGo recommended to them. However Member3 is the marginal member of this group. He mostly has a negative opinion about the items in the list but this is a main side effect of the satisfaction function which we used in BoRGo.

Table 4.10: Recommendation list of second group and member responses to the items.

Book Title	Response of Member1	Response of Member2	Response of Member3	Response of Member4	Response of Group
Lord of Chaos	Positive	Positive	Positive	Positive	Positive
The Path of Daggers	Positive	Positive	Negative	Positive	Positive
Lost Symbol	Positive	Positive	Positive	Positive	Positive
The Dragon Reborn	Positive	Positive	Negative	Positive	Positive
A Crown of Swords	Positive	Positive	Negative	Positive	Positive
The Shadow Rising	Positive	Positive	Positive	Positive	Positive
The Fires of Heaven	Positive	Positive	Negative	Positive	Positive
The Great Hunt	Positive	Positive	Positive	Positive	Positive
Knife of Dreams	Positive	Positive	Negative	Positive	Positive

(Table4.11) shows recommendation list of the third group and responses of group members to each item in the list. There are two books which are found to be negative by the third group after all of the members identified their opinions about the books in the group recommendation list.

Table 4.11: Recommendation list of third group and member responses to the items.

Book Title	Response of Member1	Response of Member2	Response of Member3	Response of Member4	Response of Group
Kuyucakli Yusuf	Positive	Negative	Negative	Positive	Positive
Binbogalar Efsanesi	Positive	Positive	Positive	Positive	Positive
Kurk Mantolu Madonna	Negative	Positive	Negative	Positive	Positive
Civisi Cikmis Dünya Uygarliklarimiz Tükendiginde	Positive	Negative	Negative	Negative	Negative
Karanligin Solugu	Negative	Positive	Positive	Positive	Positive
La Ligne Noir	Negative	Negative	Positive	Negative	Negative
Le Concile de pierre	Negative	Positive	Positive	Negative	Positive
serenad	Positive	Negative	Positive	Negative	Positive
Bu Iste Bir Yalnizlik Var	Negative	Positive	Positive	Positive	Positive

Then moderator of the group asked for the new recommendation instead of those two books -”La Ligne Noir” and ”Civisi Cikmis Dünya Uygarliklarimiz Tükendiginde”. BoRGo recommended the list in the (Table4.12). As it can be seen from the table all of the group members accepted this version of the list.

Table 4.12: Recommendation list of third group after partial recommendation.

Book Title	Response of Member1	Response of Member2	Response of Member3	Response of Member4	Response of Group
Kuyucakli Yusuf	Positive	Negative	Negative	Positive	Positive
Binbogalar Efsanesi	Positive	Positive	Positive	Positive	Positive
Kurk Mantolu Madonna	Negative	Positive	Negative	Positive	Positive
Karanligin Solugu	Negative	Positive	Positive	Positive	Positive
Le Concile de pierre	Negative	Positive	Positive	Negative	Positive
serenad	Positive	Negative	Positive	Negative	Positive
Bu Iste Bir Yalnizlik Var	Negative	Positive	Positive	Positive	Positive
Allah'in kizlari	Positive	NC	Positive	Negative	Positive
Allah Benim En el Hak	Positive	NC	Positive	Negative	Positive

In this experiment for second group, we achieved a hundred percent success. All the books we recommended have been accepted by the group. For first and third groups, eight out of nine recommended books were accepted by the group members at first. So we achieved a 0.93 average success rate without partial recommendation.

First group accepted second list which BoRGo recommended with partial recommendation. Second group accepted the first version of the recommendation list so there was no need for partial recommendation. Finally the third group also accepted second list which BoRGo recommended with partial recommendation. So it can be said that with the mediator of the BoRGo, reading groups reach a consensus over the book list to read during the year in at most two recommendation cycles over the web.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this thesis a content-based group recommender for reading groups -named BoRGo- is presented. Reading groups domain is a new domain for group recommender studies. This is the major motivation of our study. We aimed to develop a recommender system which can help reading groups to choose most acceptable books.

There is another group recommender which recommends books to its users [41].It differs from our study in the following aspects: literature reading groups are not the main focus of that study, their focus is improving the satisfaction of individuals in the group as well as the group's satisfaction. They use a two phased recommendation procedure. In first phase they use collaborative filtering method. It selects the most frequently purchased products to generate a recommendation set for a group in the first phase. In second phase of procedure they aim to reduce dissatisfied users through individual profile-based filtering. In this second phase they use the content-based approach to evaluate the compatibility between feature-based user profiles and book profiles. It is a domain independent group recommender and they used academic books as their recommendation items to evaluate their system. Aggregation method and satisfaction function which they used in that study are also different from BoRGo. Additionally they did not address after recommendation process in their study. Therefore it is not comparable with BoRGo.

BoRGo can generate recommendations for groups as well as individual users. In group recommendation process BoRGo first produces a recommendation for each member of the group. For individual recommendation process we developed two different versions. Second version's filtering method is innovative in using negative ratings of users in a negative manner in their profile. It uses the difference of positive and negative weights of a feature as its weight in filtering. We also consider only the books which have at least one common feature with the books that were rated positively by the user, as candidate books while filtering. After individual recommendation producing process BoRGo combines individual recommendation lists into a joint recommendation list for the group. This combination process needs a satisfaction function. BoRGo uses ”Average Strategy” as its satisfaction function.

After recommendation process is one of the least studied areas of recommender systems. Some other group recommenders like PolyLens [32], Pocket Restaurant Finder [33] and [55] addresses after recommendation process but none of them supply a media where members of the group can share their opinions about recommended items and ask for partial recommendation. BoRGo presents a mediator media for after recommendation process in third version. Group members can notify their opinions about each item in group recommendation list. They can also see each other's preferences about the items in the recommendation list. Moderators of the groups can ask for the partial recommendation for the items in the recommendation list which are not liked by the members in general. This is another contribution of our work.

We evaluated each of the three versions of BoRGo separately. We used most common evaluation metrics in our experiments like precision, recall , accuracy and f-measure. User studies which we performed showed 0.91 average success rate for six different groups. We saw that BoRGo successfully find relevant items for synthetic user groups which we formed due to their demographic information. We also saw that precision values remain same as the member counts of groups get larger, but recall, accuracy and f-measure values decrease. Precision values are rewarding for groups whose ratings are dense but again recall, accuracy and f-measure values are low for those kinds of groups.

When we think about what can be done as a future work after our study, there comes a few issues into mind. BoRGo is a content based recommender therefore it suffers from drawbacks of this method. In order to overcome overspecialization problem, a randomness option can be presented to moderators of the groups. For example when the moderator chooses this option, last two items of the recommendation list can be chosen randomly among the books in the database. Also thresholds can be used while finding candidate books for recommendation in order to make more precise recommendations.

Reading groups may need some filters while choosing the books which they will read during the year. BoRGo can present some of the books in recommendation list according to these filters. For example a reading group may need to read only the books which are written in Turkish, or a reading group may decide to read only poems during a year. In order to fill this kind of needs, filters can be used.

For after recommendation processes, members of the group may need a messaging facility or they may need to express their thoughts about others' choices and edits with some small notes. BoRGo may supply this kind of property to its users in the future.

We did not test the second filtering method with different kinds of items other than books. However, BoRGo can recommend any kind of item of which content information can be defined by features and dimensions, to its users. In future some experiments can be done with the filtering method in order to see the performance of it in other domains like movies, vacations etc.

We wanted to compare our system with another recommender system by using the same book dataset. However, it is theoretically possible but practically hard to adapt the available open-source recommender systems for using the same dataset with us.

REFERENCES

[1] Amazon: http://www.amazon.com/, accessed at 08.10.2011

[2] Ebay: http://www.ebay.com/, accessed at 08.10.2011

[3] Youtube: http://www.youtube.com/, accessed at 08.10.2011

[4] Google: http://www.google.com.tr/, accessed at 08.10.2011

[5] Hill W., Stead L., Rosenstein M. , and Furnas G., ”Recommending and Evaluating Choices in a Virtual Community of Use,” Proc. Conf. Human Factors in Computing Systems, 1995.

[6] Resnick P., Iakovou N., Sushak M., Bergstrom P. , and Riedl J., ”GroupLens: An Open Architecture for Collaborative Filtering of Netnews,” Proc. 1994 Computer Supported Cooperative Work Conf., 1994.

[7] Shardanand U. and Maes P., ”Social Information Filtering: Algorithms for Automating ’Word of Mouth’,” Proc. Conf. Human Factors in Computing Systems, 1995.

[8] Rich E., ”User Modeling via Stereotypes,” Cognitive Science, vol. 3, no. 4, pp. 329 354, 1979.

[9] Powell M.J.D., Approximation Theory and Methods. Cambridge Univ. Press, 1981.

[10] Salton G., Automatic Text Processing. Addison Wesley, 1989.

[11] Armstrong J.S., Principles of Forecasting A Handbook for Researchers and Practitioners. Kluwer Academic, 2001.

[12] Yates R. B., and Neto B. R., Modern Information Retrieval. Addison Wesley, 1999.

[13] Lilien G.L., Kotler P., and Moorthy K.S., Marketing Models. Prentice Hall, 1992.

[14] Adomavicius G., Tuzhilin A. ”Toward the next generation of recommender systems: A survey of the state of the art and possible extensions”, IEEE Transactions on Knowledge and Data Engineering 17, 734 749, June 2005.

[15] Rich E., ”User Modeling via Stereotypes,” Cognitive Science, vol. 3, no. 4, pp. 329 354, 1979.

[16] Burke R. Hybrid recommender systems: Survey and experiments. User Modeling and User Adapted Interaction, 12(4):331 370, 2002.

[17] Peis E., Castello J.M.M., DelgadoLopez J.A., 2008. Semantic Recommender Systems: analysis of the state of the topic. In Hypertext, (2008).

[18] Goldberg D., Nichols D., Oki B.M., and Terry D., ”Using Collaborative Filtering to Weave an Information Tapestry,” Comm. ACM, vol. 35, no. 12, pp. 61 70, 1992.

[19] Breese J.S., Heckerman D., and Kadie C., ”Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” Proc. 14th Conf. Uncertainty in Artificial Intelligence, July 1998.

[20] Wang R.Q. and Kong F.S., Semantic enhanced personalized recommender system. 2007 International Conference on Machine Learning and Cybernetics, (August):4069 4074, 2007.

[21] Ozsoy M., Cicekli I., Alpaslan F., 2010. Text Summarization of Turkish Texts using Latent Semantic Analysis. In Proceedings of of the 23rd International Conference on Computational Linguistics (Beijing, China, 2010).

[22] Mooney R.J., Bennett P.N., and Roy L., ”Book Recommending Using Text Categorization with Extracted Information” Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS 98 08, 1998.

[23] Pazzani M. and Billsus D., ”Learning and Revising User Profiles: The Identification of Interesting Web Sites” Machine Learning, vol. 27, pp. 313 331, 1997.

[24] Adomavicius G. and Tuzhilin A., ”Multidimensional Recommender Systems: A Data Warehousing Approach,” Proc. Second Intl Workshop Electronic Commerce (WELCOM 01), 2001b.

[25] Adomavicius G., Sankaranarayanan R., Sen S., and Tuzhilin A., ”Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach,” ACM Trans. Information Systems, vol. 23, no. 1, Jan. 2005

[26] LIU H. AND MOTODA H. 1998. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers.

[27] CHATTERJEE S., HADI A. S., AND PRICE B. 2000. Regression Analysis by Example. John Wiley and Sons, Inc.

[28] KIMBALL R. 1996. The Data Warehouse Toolkit. John Wiley & Sons, Inc.

[29] CHAUDHURI S. AND DAYAL U. 1997. An overview of data warehousing and OLAP technology. ACM SIGMOD Record 26, 1, 65 74.

[30] Joseph F. Mccarthy, Theodore D. Anagnost, 1998. MusicFX: An arbiter of group preferences for computer supported collaborative workouts. In: Proceedings of the 1998 Conference on ComputerSupported Cooperative Work. (1998) 363 372

[31] Hill W., Stead L., Rosenstein M., and Furnas G. (1995): Recommending and Evaluating Choices in a Virtual Community of Use, in Conference Proceedings on Human Factors in Computing Systems, Denver, CO, 1995, pp. 194201.

[32] O’Connor M. , Cosley D., Konstan J. A. and Riedl J., 2001. PolyLens: A recommender system for groups of users. In Prinz,W., Jarke, M., Rogers, Y., Schmidt, K.,Wulf, V., eds.: Proceedings of the Seventh European Conference

on ComputerSupported Cooperative Work. Kluwer, Dordrecht, The Netherlands (2001)

[33] Mccarthy J. F., 2002. Pocket Restaurant Finder A situated recommender system for groups. In Proceedings of theWorkshop on Mobile Ad Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems, Minneapolis.

[34] Ardissono L., Goy A., Petrone G., Segnan M., Torasso P., 2003. INTRIGUE Personalized recommendation of tourist attractions for desktop and handset devices. Applied Artificial Intelligence 17(8 9) (2003) 687 714

[35] Jameson, A. 2004. More than the sum of its members: Challenges for group recommender systems. In Proceedings of the International Working Conference on Advanced Visual Interfaces, 4854

[36] Yu Z., Zhou X., and Zhang D., An Adaptive InVehicle Multimedia Recommender for Group Users, IEEE 61st Semiannual Vehicular Technology Conference (VTC2005Spring), pp. 28002804, IEEE Press, May 30 June 1, 2005, Clarion Hotel, Stockholm, Sweden.

[37] McCarthy K., Salam? M., Coyle L., McGinty L., Smyth B., Nixon P., 2006. CATS: A Synchronous Approach to Collaborative Group Recommendation. FLAIRS Conference 2006: 8691

[38] McGinty L., and Smyth B. 2003. Tweaking Critiquing. In Proceedings of the Workshop on Personalization and Web Techniques at the International Joint Conference on Artificial Intelligence. MorganKaufmann.

[39] Burke R., Hammond K., and Kozlovsky J. 1995. Knowledge based Information Retrieval for Semi Structured Text. In Proceedings of the AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval, 19 24. AAAI Press.

[40] Yu Z., Zhou X., Hao Y., and Gu J., TV Program Recommendation for Multiple Viewers Based on User Profile Merging, User Modeling and User -Adapted Interaction (UMUAI), Springer Press, Vol. 16, No. 1, March 2006, pp. 63-82.

[41] Kim J. K., Kim H. K., Oh H. Y., Ryu Y. U., 2009. A group recommendation system for online communities. In: International Journal of Information management (2009) Volume: 30, Issue: 3, Publisher: Elsevier Ltd, Pages: 212-219 ISSN: 02684012 DOI: 10.1016 j.ijinfomgt.2009.09.006

[42] Garcia I., Sebastia L., Onaindia E., Guzman C., A Group Recommender System for Tourist Activities, Proceedings of the 10th International Conference on E-Commerce and Web Technologies, September 01-04, 2009, Linz, Austria

[43] Condorcet M. de (1785). Essai sur l'application de l'analyse a' la probalite des decisions rendues a' la pluralite des voix. Paris.

[44] Pattanaik P.K. (1971). Voting and CollectiveChoice. Cambridge: CambridgeUniversity Press.

[45] Taylor A. (1995) Mathematics and politics: Strategy, voting, power and proof. New York: Springer Verlag.

[46] Cohen W., Schapire R. and Singer Y. (1999). Learning to order things. Journal of Artificial Intelligence Research, 10, 243-270.

[47] Dwork C., Kumar R., Naor M. and Sivakumar D. (2001). Rank aggregation methods for the web. Tenth International World Wide Web Conference, Hong Kong, pp. 613622.

[48] Fagin R., Lotem A. and Naor M. (2003). Optimal aggregation algorithms for middleware. Journal of Computing System Sciences, 66, 614-656.

[49] Pennock D., Horvitz E. and Giles C. L. (2000). Social choice theory and recommender Systems: Analysis of the axiomatic foundations of collaborative filtering. 17th National Conference on Artificial Intelligence, Austin, TX, pp. 729-734

[50] Hogg L. and Jennings N.R. (1999). Variable sociability in agent-based decision making. Sixth International Workshop on Agent Theories Architectures and Languages, Orlando, FL, USA, pp. 276-289

[51] Ephrati E. and Rosenschein, J.S. (1996). Deriving consensus in multi agent systems. Artificial Intelligence, 87, 21-74

[52] Masthoff J., 2004. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. In : User Modeling and User -Adapted Interaction, Vol. 14, No. 1. (1 February 2004), pp. 37-85.

[53] Borda J.C. (1781). Memoire sur les elections au scrutine. Histoire de l'Academie Royale des Sciences.

[54] Copeland A.H.(1951).AReasonableSocialWelfareFunction.Mimeo,University of Michigan.

[55] Jameson A., Baldes St., and Kleinbauer T.: Enhancing Mutual Awareness in Group Recommender Systems (2003) In B. Mobasher , S. S. Anand (Eds.), Proceedings of the IJCAI 2003 Workshop on Intelligent Techniques for Web Personalization, Menlo Park, CA, USA

[56] Reading Group Choices: http://www.readinggroupchoices.com/readinggroups/choose.cfm. 2011.

[57] Thyke: http://www.thyke.com/, 10.11.2011.

[58] Igeder: http://www.igeder.org.tr/sayfa.php?id=116, 10.11.2011.

[59] Bilim ve Sanat Vakfi: http://www.bisav.org.tr, 10.11.2011.

[60] Berkovsky S., Kuflik T., Ricci F., "Cross-Technique Mediation of User Models", In proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, (AH), Dublin, Ireland, 2006.

[61] Perugini S., Gonçalves M. A. and Fox E. A. , Recommender Systems Research: A Connection-Centric Survey, Journal of Intelligent Information Systems Volume 23, Number 2, 107-143, DOI: 10.1023/B:JIIS.0000039532.05533.99

[62] UserDataset : http://www.informatik.uni-freiburg.de/ cziegler/BX/, Last accessed on December 2010.

[63] BookDataset : http://infochimps.com/datasets/book, Last accessed on January 2011.

[64] Sebastiani F., Machine learning in automated text categorization, ACM Computing Surveys (CSUR), v.34 n.1, p.1-47, March 2002 [doi¿10.1145/505282.505283]

[65] KIRMEMIS O., BIRTURK A., A Content-Based User Model Generation and Optimization Approach for Movie Recommendation, In 6th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems at the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-08), 13-17 July 2008, Chicago, Illinois.

[66] KIRMEMIS O., BIRTURK A., OPENMORE: A User Controlled Content Based Movie Recommender with Explanation and Negative Feedback, In 4th International Conference on Web Information Systems and Technologies (WEBIST '08), 4-7 May 2008, Funchal, Portugal.

[67] KIRMEMIS O., BIRTURK A., A Content Based Movie Recommendation System, in Expanding the Knowledge Economy: Issues, Applications, Case Studies, Paul Cunningham and Miriam Cunningham(Eds), IOS Press, 2007 Amsterdam, ISBN 978-1-58603-801-4; eChallenges 2007, 24-26 October 2007, The Hague, The Netherlands.

[68] Billsus D., Pazzani M.J., "Learning collaborative information filters", In Proceedings of the 15th National Conference on Artificial Intelligence, 1998.

[69] Basu C., Hirsh H., Cohen W., "Recommendation as Classification: Using Social and Content-Based Information in Recommendation", Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08, AAAI Press, 1998.

[70] Sarwar B., Karypis G., Konstan J., Riedl J., "Analysis of recommendation algorithms fore-commerce", In Proceedings of ACM E-Commerce, 2000.

[71] What is ISBN: http://ussan.org/standards/home/isbn/us/isbnqa.asp, at 01.12.2011

[72] Facebook: www.facebook.com, accessed at 01.12.2011

[73] Baltrunas L., Makcinskas T., and Ricci F., 2010. Group recommendations with rank aggregation and collaborative filtering. In Proceedings of the fourth ACM conference on Recommender systems (RecSys '10). ACM, New York, NY, USA, 119-126.

[74] Berkovsky S. and Freyne J., 2010. Group-based recipe recommendations: analysis of data aggregation strategies. In Proceedings of the fourth ACM conference on Recommender systems (RecSys '10). ACM, New York, NY, USA, 111-118

[75] Css-template: http://www.webdesignlondon-tristar.co.uk/, accessed at 30.12.2011

APPENDIX A

PUBLICATIONS

Sayil Duzgun, Aysenur Birturk, BoRGo: A Book Recommender for Reading Groups, ACM RecSys'11 Workshop on ”Human Decision Making in Recommender Systems”, In conjunction with the 5th ACM Conference on Recommender Systems, 22-27 October 2011, Chicago, Illinois, USA.