ABSTRACT

COLLABORATIVE FILTERING IN TV RECOMMENDER

by
Elizabeth Podberezniak

The thesis describes different types of collaborative filtering methods to filter information from the large amount available and presents examples of such systems in different domains. It focuses on automated collaborative filtering to generate personalized recommendation of information.

Different variations of the automated collaborative filtering scheme are developed and analyzed in the thesis. An additional adjustment of the predicted score is implemented in order to improve precision of the recommendation. Different combinations of parameters are analyzed to maximize system effectiveness.

The data for the analysis was gathered through TV Recommender, a World Wide Web system developed for the thesis. The TV Recommender is a fully functional system that acquires users' data and implements the enhanced collaborative filtering scheme to generate user's personalized TV recommendation.

COLLABORATIVE FILTERING IN TV RECOMMENDER

by
Elizabeth Podberezniak

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

May 1998

APPROVAL PAGE

COLLABORATIVE FILTERING IN TV RECOMMENDER

Elizabeth Podberezniak

Dr. John D. Carpinelli, Thesis Advisor — Date
Associate Professor of Electrical and Computer Engineering and Computer and Information Sciences, NJIT

Dr. Timothy Chang, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Walid Hubbi, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Elizabeth Podberezniak

Degree: Master of Science in Electrical Engineering

Date: May 1998

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, 1998

- Bachelor of Science in Computer Engineering,
  New Jersey Institute of Technology, 1992

Major: Electrical Engineering

This thesis is dedicated to my parents
Olga and Dymitr Podberezniak

ACKNOWLEDGMENTS

The author wishes to express her sincere gratitude to her advisor, Professor John D. Carpinelli for his encouragement, guidance, friendship and support throughout this research and throughout her time at NJIT.

Special thanks are included to Professors Timothy Chang and Walid Hubbi for serving as members of the committee.

The author is grateful to Professor Ali N. Akansu, NJIT ECE Dept and Co-Director for making New Jersey Center for Multimedia Research, NJCMR facility, available for development and analysis of the research.

The author appreciates help and suggestions from users of TV Recommender, especially Michael Chinni and many others, who provided significant feedback.

The author wishes to thank her family for understanding and constant encouragement to complete the graduate study. Without their support, it would not be possible.

And finally, the author wishes to thank the members of Electrical and Computer Department at New Jersey Institute of Technology and the members of Human Resource Technology Group at Coopers and Lybrand for their support.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 The need of Information Filtering

In today's age the amount of information is enormous. With the increase of computer power and connectivity the access to information for an average user has increased even more. However, the ease for finding relevant information about a specific subject matter becomes more and more difficult. The user is faced with a list of catalogs and articles to search and sort checking for relevance. Searching for more personal information such as restaurants to visit, television programs to watch or music to listen are merely impossible.

Today it is not a time when the person was able to research a subject, analyze and choose the one appropriate. Because of the large amount of information, a user can not possibly find all available information, sort through and pick the best candidates in a reasonable amount of time.

Society today is aware of the problem of choosing information. This is why we have top ten lists of movies, music albums, etc. Those lists are composed based on a rating of a group of people. There are also lists of the most interesting items created by experts or critics. All of that is useful, but what about the individuals that do not necessarily agree with the expert's options or average opinion of some group of people? In one subject matter or another all of us do not always agree with the ratings provided. There is a great need for more personalized information filtering specific to the individuality of all of us[1].

1.2 Method of Information Filtering

1.2.1 Content Filtering

The content information filtering is a keyword-based filtering. If a specific keyword exists in the document in the database that document would be recommended[1]. It is a very simple and practical search engine. However there are a number of disadvantages to the content-based filtering:

- It is valid for documents only, where the machine can parse and analyze each word. The content-based filtering can not understand graphics, audio or video files, which become more and more common.
- It will make recommendations based on some analytical criteria, which may be the number of times the keyword is used in the document, but it knows nothing about the quality of the document.
- It does not provide a user with a chance to find new documents and keywords that may be related, but not specified by the user. The user can not discover items that he/she is not aware of already.

1.2.2 Active Collaborative Filtering

Collaborative filtering relies on a human intelligence to recommend items, not some analytically calculated criteria as in content-based filtering. It is known as a "word of mouth" filtering. Normally when we seek information, we would talk to friends and ask experts and based on their opinions make our own recommendations.

Collaborative Filtering is called active when the user must identify its close friends with whom he/she shares interests and opinions. Based on the user's interests and ratings of the friends, the system would calculate recommendations for the user[2]. The active collaborative filtering systems can also be further divided based on the direction of information travel. If the user specifies people and subjects of the close opinions to be able to formulate his/her own recommendation, it is called Pull Active Collaborative Filtering. When the user designated other users that may be interested in his/her opinions, it is known as Push Active Collaborative Filtering[1].

Active Collaborative Filtering, although more personal than content-based filtering, still faces disadvantages. One of the main ones is that the user must identify other users with similar interests in a given subject matter. It is possible only within a small group of people where everyone knows everyone else and their expertise. If that is the case, we are back to the drawbacks of the mouth to mouth sharing information, just adding all the computer complexity.

1.2.3 Automated Collaborative Filtering

Automated Collaborative Filtering, in addition to determining recommendations while users with close interests are provided, can also specify the individuals who share the user's interests and opinions. The system is able to determine the closeness of the user to all other users, choose the ones that are the closest according to their interests and opinions, and leverage their collective ratings in order to determine recommendation for the user[1]. Just as a user would have to ask other users for their opinion, the automated collaborative filtering system finds other users to determine recommendations. The system, having a large number of users, can find more people with whom the user would share interests, and with greater precision determine the recommendation by leveraging all ratings of interest sharing friends.

Automated Collaborative Filtering has many advantages over the content-based filtering and the active collaborative filtering:

- It is human intelligence driven; it performs recommendations for all types of items, not just documents where the data needs to be parsed and keywords searched.
- It uses human, rather then machine intelligence; it can make highly subjective recommendations based on people's likes and dislikes.
- It relies on the fact that people's opinions about specific items are not randomly distributed and is able to determine the pattern and use it in determining recommendation.
- It does not require the user to know people with close interests and opinions.
- It enables the user to discover new items that the user was not aware of by examining people whose interests are similar to those to the user.

It is worth noting however that the Automated Collaborative Filtering is very efficient for systems where the domain is very highly subjective (music, television programs). In a case of a widespread domain (web pages, books) it may provide inaccurate recommendations. It is due to the fact that the domain may have a number of sub-domains and people's opinions may vary greatly for each sub-domain. If the sub-domains are not recognized, the recommendation may not be accurate.

1.2.4 Feature-Guided Collaborative Filtering

The Feature-Guided Collaborative Filtering acknowledges the potential problem and partitions the domain into a number of sub-domains and performs a standard Automated Collaborative Filtering for each domain. In fact, the Feature-Guided Automated Collaborative Filtering takes the advantage of content-based filtering and combines it with all the advantages of standard Automatic Collaborative Filtering[1].

1.3 Outline of the Document

The rest of this thesis is organized as follows. Chapter 2 contains examples of existing collaborative filtering systems. Chapter 3 presents TV Recommender, a World Wide Web system for recommending television programs developed for the thesis. It describes the user interface and technical details of the recommendation engine. Chapter 4 presents analysis of the system including results and conclusions. Qualitative results are also included in the chapter. Chapter 5 details possible future research on TV Recommender and collaborative filtering in general.

CHAPTER 2

EXAMPLES OF COLLABORATIVE SYSTEMS

There are few collaborative filtering systems available today supplying recommendations for different items in music, movies, books, web sites, newsgroups, restaurants, etc. A number of universities and corporations are involved in research of providing adequate information to a user. Examples are included below.

2.1 Tapestry

Tapestry is one of the first collaborative filtering systems for recommending electronic documents via e-mail or Netnews developed at XEROX PARC. It is an example of an active collaborative filtering system. A user is able to retrieve documents not only by content, but also based on ratings of other users. However, in that system the user must specify the other users with whom he/she would share interests to be able to obtain recommendations[3].

2.2 GroupLens

GroupLens system is part of the ongoing GroupLens research project at the University of Minnesota Department of Computer Science. It is a system recommending articles for Usenet news. After a user finishes reading an article, he/she is asked to rate it on a scale 1 to 5, based on how interesting he/she found the article. The agreement degree between two users is obtained based on articles that both users have read and rated. A prediction of a given article is calculated by taking all ratings for that article and weighting them by the correlation between the rater and the user requesting the prediction. Whenever there is a new article, the system uses the ratings of other people who agree with the user to generate a prediction of his/her interests and content of the article[4].

The protocol for asking for a prediction consists of filling out a table with the user id and article ids to predict. During the collection of data, the system also collects the amount of time spent reading the article, the number of lines and characters in the article and also detects if the user applied any special actions after reading the article, such as forward, save, etc. The GroupLens system is able to rate the system based on the time spent reading the article and the number of lines of the article even if the user did not bother to rate the article. Also if the user saves the article or forwards it, that is also an indication of a rate to the GroupLens system[5].

2.3 Firefly

Firefly is one of the more popular collaborative filtering systems carrying a wide range of different items to recommend and predict. It was developed by Firefly Networks, Inc. based on a number of projects researched at MIT, with a music recommending system RINGO as a pioneer. The Firefly product is designed based on Automated Collaborative Filtering technology and Feature-Guided Automated Collaborative Filtering technology and can by implemented on a number of different items. It consists of three core products: the Firefly Passport Office, Community Navigator and the Catalog Navigator.

The Firefly Passport Office is a central profile management unit that registers new users, recognizes existing users, tracks any profile changes, personalizes sites visited by the user and builds communities. The user is able to see other users visiting the same site at the same time and exchange information between the users. It allows personalization of web sites appearance based on demographics of the visiting user. It is also able to generate reports of traffic of specified sites.

The Firefly Community Navigator applies Automated Collaborative Filtering technology to build communities of the same likes and dislikes. It can also inform users of like-minded friends. The Firefly Catalog Navigator uses Automated Collaborative Filtering to let users intelligently navigate through any catalog available on a site. It recommends items based on people's tastes. In the case of large domain catalogs, it applies the Feature-Guided Automated Collaborative Filtering methodology to subdivide the large domain and recommend items within the subdivision[6].

CHAPTER 3

DESCRIPTION OF TV RECOMMENDER

3.1 System Overview

The TV Recommender is a system for recommending TV programs to users after learning their TV program tastes. For some of the system web screens refer to Appendix A. Help files are included in Appendix B. The TV Recommender introduces an Adjusted Automated Collaborative Filtering algorithm to generate recommendation. It asks a user to rate a number of TV programs and based on those rates and on rates of other people with similar tastes it creates a prediction. In addition, based on average user's rate in a given program category, it is able to recommend which other programs the user would like to watch and which programs he/she should avoid. The features of TV Recommender are as follows.

Personal choices:

- provides personalized top 20 suggested programs that the user would like to watch
- provides personalized top 20 suggested programs that the user would not like to watch
- makes a prediction about a specific TV program
- allows the user to add specific program rate to the user's profile
- provides user's profile – lists programs that the user rated and their rates

Generic choices:

- provides general top 20 most liked programs and their average rates
- provides general top 20 most popular programs and number of people that rated them
- provides a list of all programs with the average rate and rated count
- allows user to add programs to database
- allows users to share (store and retrieve) comments about each of the programs

The main challenge of the project is the personalized choice ability to generate recommendations of programs to watch and programs to avoid watching. The generation of the prediction process can be divided into three parts: Data Collection, Neighborhood Builder and Recommendation Engine. The system overview is presented in Figure 1.

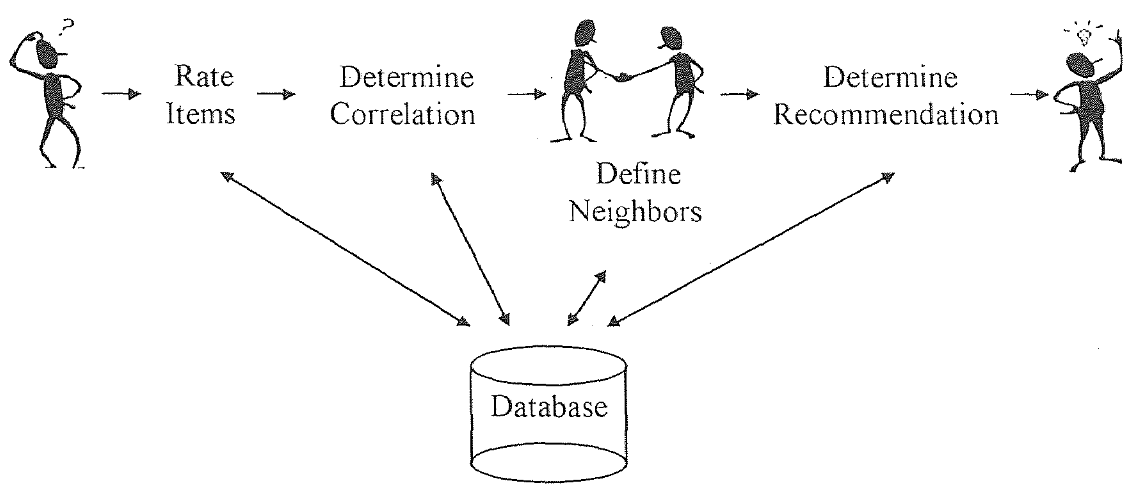

Figure 1 The TV Recommender System Overview

3.2 Data Collection

When a user enters the TV Recommender site, he/she can enter the system as a New User if it is his/her first time or Login. A New User is asked to enter some personal data, which for TV Recommender consists of name, password and an email address. If the user omits some data or the name already exists, the user is prompted for a new set of personal data. If a user logs into the system using the Login button and his/her data is already in a system, he/she is prompted for a name and password. After a successful entry to the system, a user is presented with 30 randomly chosen TV programs and asked to rates at least 20 of them. If the user is not able to rate 20 programs from the 30 presented, he/she can submit whichever programs he/she is able to rate and the system will display another 30 randomly chosen programs remembering those already rated. It also presents a counter of already rated programs. The user can submit the ratings multiple times until he/she successfully completes ratings for at least 20 TV programs. Only after the user provides ratings for at least 20 programs, his/her personal data and rating scores are stored in the system database. The TV Recommender treats personal data entry and TV program rating collection as one task because, unless the user rated a sufficient number of TV programs, the system is not able to generate the recommendation. Therefore, users who did not rate the minimum number of programs are not recognized by the system and not registered.

The rating scale is presented in Figure 2.

0 – Don't Know

1 - Hate It

2 - Pretty Bad

3 - Not My Taste

4 - It's OK

5 - I like It

6 - Great Stuff

7 - The Best

Figure 2 Rating Scale

The user is presented with the verbal phrases to rate the programs. The number next to each phrase is a system's interpretation for each rate. The scale is absolute, not normalized. Users have different rating styles: some people rate only programs that they like, some other only programs that hate, therefore an absolute rating scale is the most appropriate.

3.3 Neighborhood Builder

Neighborhood Builder part of the system is responsible for defining a neighborhood of people with similar tastes for TV programs. It is based on the constrained Pearson r algorithm used in the prototypes of Firefly, music recommending system Ringo, which determines a closeness of two sets of data[7].

The difference between the traditional Pearson r algorithm and the constrained correlation coefficient algorithm is that the values are normalized against the middle value of 4.

$$C_{xy} = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}} \tag{3.1}$$

where:

$$S_{xy} = \sum_{i=1}^{n}\left((x_i - A)(y_i - A)\right) - \frac{\left(\sum_{i=1}^{n}(x_i - A)\right)\left(\sum_{i=1}^{n}(y_i - A)\right)}{n} \tag{3.2}$$

$$S_{xx} = \sum_{i=1}^{n}(x_i - A)^2 - \frac{\left(\sum_{i=1}^{n}(x_i - A)\right)^2}{n} \tag{3.3}$$

$$S_{yy} = \sum_{i=1}^{n}(y_i - A)^2 - \frac{\left(\sum_{i=1}^{n}(y_i - A)\right)^2}{n} \tag{3.4}$$

where:

C_{xy} - correlation coefficient

x_i - score given by prediction requester i

y_i - score given by user i in a database

A - average rate = 4

n - number of items in a database

with $-1 < C_{xy} < 1$

A C_{xy} of value 0 indicates no correlation. The greater the value of C_{xy}, the greater there is a similarity between the two sets of data. For not rated programs an average score of 4 is assumed.

Due to the extended calculations required to obtain correlation coefficients, they are done only once in a case of a new user entering the system or of a user changing ratings in his/her profile stored in a database.

Knowing the closeness degree between the user and all other users in the database, the system is able to determine neighborhood of users of similar TV tastes by including all users whose correlation coefficient C_{xy} is greater than a certain threshold value T_1.

3.4 Recommendation Engine

Recommendation Engine in TV Recommender is responsible for calculating predictions for specific programs. It is done in the following steps.

3.4.1 Weight Calculation

Calculate the weight for each user in the neighborhood with respect to the prediction requester or guest.

$$W_{xy} = \left(\frac{C_{xy} - T_1}{1 - T_1}\right)^2 \text{ for } C_{xy} >= 0 \quad (3.5)$$

$$W_{xy} = -\left(\frac{-C_{xy} + T_1}{1 + T_1}\right)^2 \text{ for } C_{xy} < 0 \quad (3.6)$$

where:

W_{xy} - weight of each user y in the neighborhood with respect to prediction requester x

C_{xy} - correlation coefficient between prediction requester x and person y from the database

T_1 - neighborhood threshold value

The weight W_{xy} is proportional to the correlation coefficient C_{xy}. Figure 3 presents the relationship between correlation coefficient C_{xy} and weight W_{xy} for different values of the threshold T_1. Note that both positive and negative values are used by the system to determine the weight. From the chart, it is noticeable that a higher absolute value of T_1 would build smaller and therefore closer neighborhood with the weight adjusted appropriately from 0 to 1. On the other hand, the absolute values of T_1 must be low enough to be able to create the neighborhood of users, which is proportional to the number of users in the database and the close relationship between users.

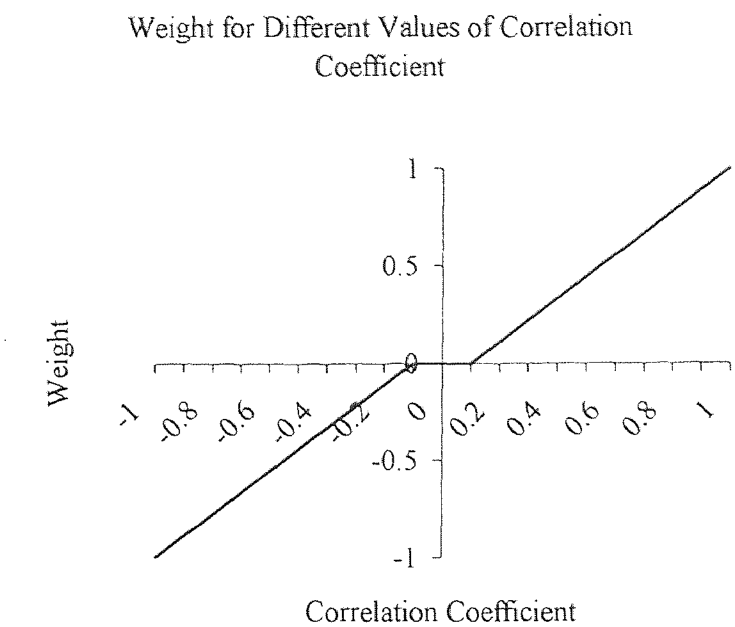

Figure 3 Weight for Different Values of Correlation Coefficient for Threshold $T_1 = 0.1$.

3.4.2 Score Prediction

The predicted score for a program is calculated as a weighted average of all ratings in the neighborhood for the item. Similar to the correlation coefficient calculations, it is also normalized around the middle value of 4.

$$P_{xp} = \frac{\sum_{i=1}^{N} \left(W_{xy} \left(R_{yp} - A \right) \right)}{\sum_{i=1}^{N} \left(W_{xy} \cdot k \right)} + A \qquad (3.7)$$

where:

P_{xp} - predicted score for user x for item p

W_{xy} - weight of prediction requester x and person y from the neighborhood

R_{yp} - rate of program p given by a user y in the neighborhood

$k = 0$ if user y did not rate program p

$k = 1$ if user y rated program p

The score is calculated from all programs that were rated by the neighbors of similar tastes, excluding the programs rated by the prediction requester.

3.4.3 Score Adjustment

Adjusted value is a predicted score adjusted based upon the average rating of the prediction requester for the recommended program category. In general the user does agree with the taste of people in his/her taste neighborhood, but he/she may not fully agree in tastes of different TV program categories. The TV Recommender internally divides all programs into specific categories presented in Figure 4.

Children TV	Comedy
Documentary/News	Drama
Game/Show	Technology/Science Fiction
Sport	Home/Leisure
Soap Opera	Movies
Talk Show	Music

Figure 4 List of Program Categories

It calculates average rate of all programs rated by prediction requester for each of the categories and uses that value to adjust the predicted score for a program. The adjusted value is calculated as:

$$P_{xp} = P_{xp}\left(T_2\left(AR_{xc} - A\right) + 1\right) \quad (3.8)$$

where:

P_{xp} - predicted score

AR_{xc} - average rate of prediction requester for all items in category c

T_2 - prediction requester threshold value

A - average rate = 4

By adding the adjustment for the predicted score, the final recommendation will depend not only on a recommendations of users with similar TV tastes but also on likes and dislikes of the prediction requester of programs in different categories.

When the calculations are done, displaying of the top 20 programs to watch or avoid watching is simply sorting the list in descending or ascending order and displaying the top 20 programs.

3.5 Database Design

The database of the TV Recommender is a set of flat files stored in a separate directory referenced by a program through a setup file. All of them are pipe-separated, variable length files. Each user is referenced in the database by a name, while each program and each program category are referenced by a system assigned index.

The TV Recommender database consists of:

- item catalog – contains TV programs data: program index, program name and the catalog index to which the program belongs
- category catalog – contains program category index and category name
- people profile – contains personal user information: user name, password and e-mail address
- people rates – contains rates of programs for each person registered
- correlation – contains values of correlation between any two people in the system
- score – contains predicted ratings for programs for each person registered
- comment – contains program index and comments about the program given by users

Each of the files above is accompanied by a log file to store the history of the file and a lock file necessary for allowing access to the file.

CHAPTER 4

ANALYSIS AND RESULTS

4.1 General Observations

The TV Recommender received mostly positive response from users of the system. Many of them classified personal recommendation as "an interesting idea." They were eager to share their opinions about their recommendations as well as about the design of the system. A sample of user comments is included in Appendix C. Originally the system provided 100 programs and for the purpose of collecting data for recommendation analysis, it did not allow the user to alter or add more programs. Once the required data was collected, the TV Recommender was enhanced with a feature to add programs to the database, which was the most common user request. Another enhancement recommended by users was the ability to view the user's profile, which was also added to the TV Recommender options menu. The programs presented to users for ratings were chosen randomly and, since the original design did not include adjusting of the predicted score, the program distribution among the categories is very uneven. The distribution is presented in Figure 5. The number of programs and the distribution of programs among different categories may effect the precision of the adjustment.

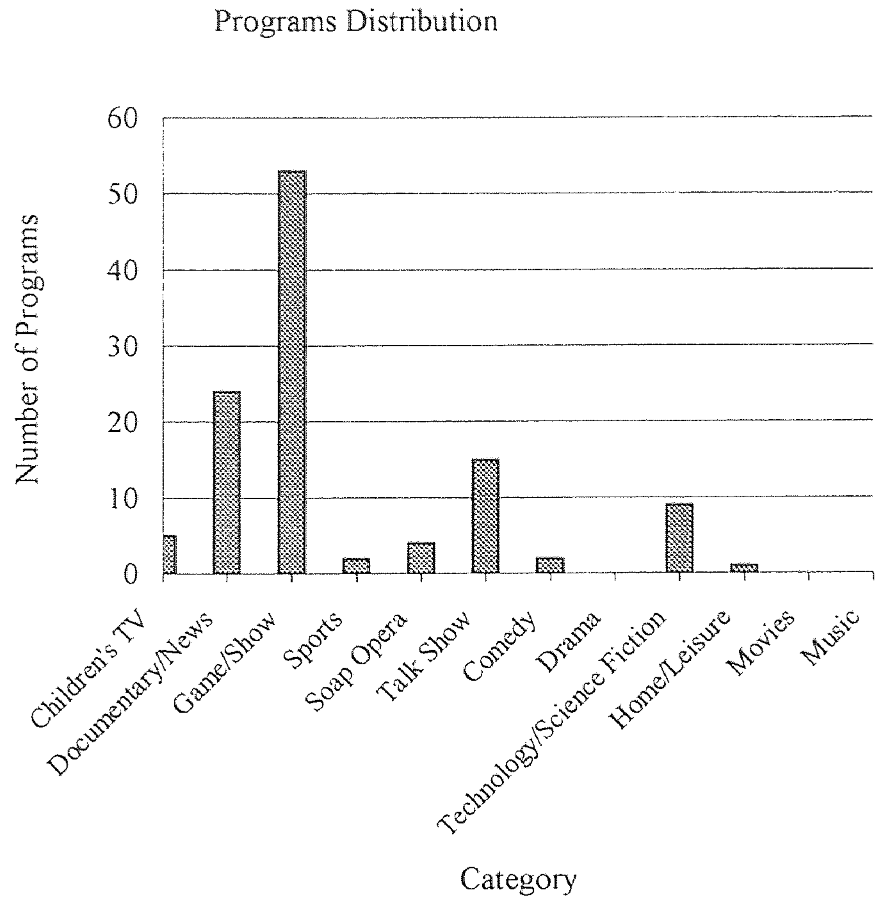

Figure 5 The Category Distribution

Users were presented with at least 30 randomly chosen programs and required to rate at least 20 programs. The least number of programs rated by a user was 20 and the maximum number rated by a single user was 75. The distribution of programs' scores in a user's profile is presented in Figure 6 and the distribution of mean scores for each profile is shown in Figure 7. Note that the user's mean scores tend to be slightly lower than the average rate of 4; however, from the responses gathered, users expected higher recommendations.

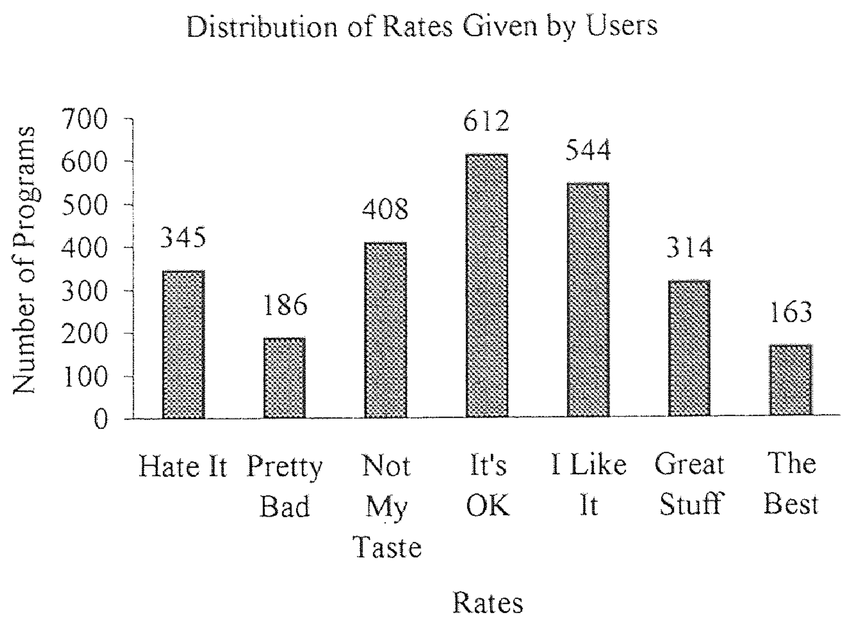

Figure 6 The Distribution of Rates Given by Users

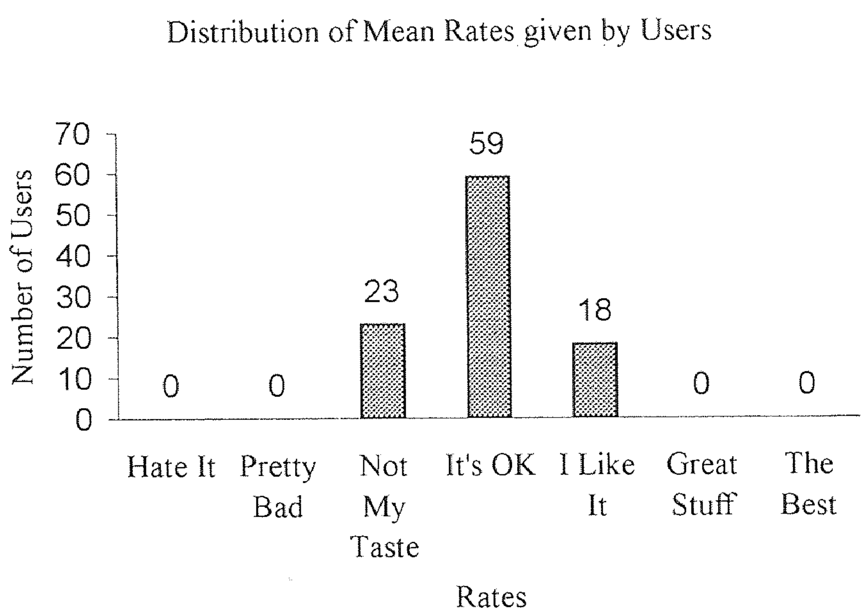

Figure 7 The Distribution of Mean Rates Given by Users

In comparison to the music recommendation system Ringo, the TV Recommender's correlation coefficients between users is lower, which is presented in Figure 8[7].

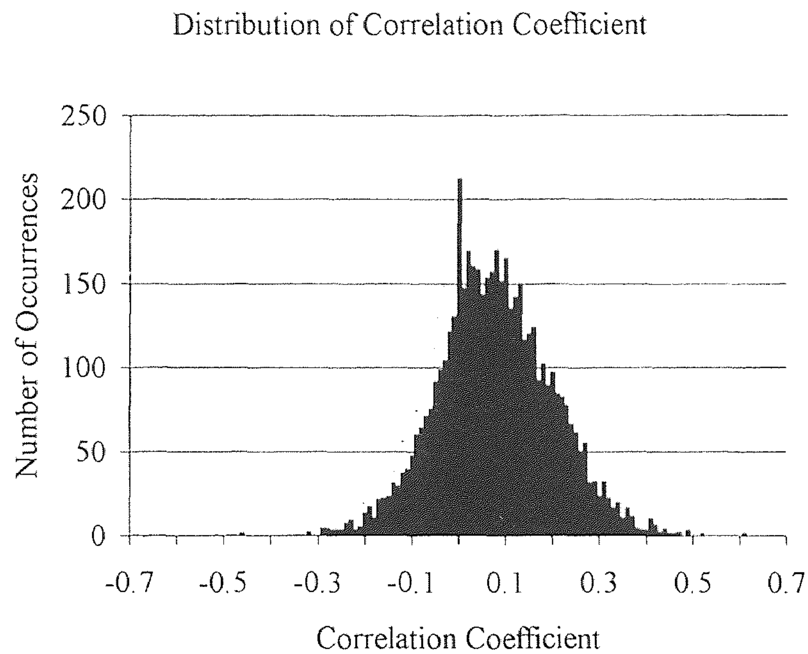

Figure 8 The Distribution of Correlation Coefficient

In general the correlation coefficient has a bell shape. It is noticeable however, that the system contains an abnormally large number of users whose correlation coefficient is 0.0, meaning that those users can not share their tastes in TV programs. The cause of the behavior is not known. Speculations can be made that it is due to the fact that users were limited to the programs in the TV Recommender database and did not have the ability to rate any other programs of their choice, which may have forced some of them to give less precise ratings.

4.2 Evaluation Criteria

The general observations were done on a full population of 100 users. To further analyze the prediction scheme, the set of 100 people was divided into a source set $\{s_1 \ldots s_n\}$ and target set $\{t_1 \ldots t_n\}$. Since the minimum programs to rate was 20, 4 randomly chosen ratings were included in the target set and the remaining 16 or more ratings were included into the source set. The recommendations $\{p_1 \ldots p_n\}$ were generated based on the source set and compared to the values in target set. The evaluation criteria were such that the mean absolute error and the standard deviation would be minimized[8]. The number of target values that the system was able to predict was taken into consideration, but it was treated with secondary importance. In general, with the large amount of information available, it seems to be more beneficial to recommend a few items precisely, rather than include many items of which the system is not certain.

The mean absolute error is calculated as a weighted average of all absolute errors:

$$|\overline{E}| = \frac{\sum_{i=1}^{N} |\varepsilon_i|}{N} \tag{4.1}$$

The standard deviation is calculated as follows:

$$\sigma = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}\left(\varepsilon_i - |\overline{E}|\right)^2\right)} \tag{4.2}$$

The mean square error is calculated as:

$$MSE = \frac{\sum_{i=1}^{N} (\varepsilon_i)^2}{N}$$

where:

$\varepsilon_i = t_i - p_i$

N - number of programs the system can predict

In order to evaluate the adjustment of the predicted value, at first the system would be analyzed without the adjusting factor for different variations of T_1 only. Then the adjustment would be added for different combinations of T_1 and T_2 in order to produce the best prediction scheme.

4.3 Analysis without Adjusting Factor

At first the recommendation algorithm was evaluated without any adjustment of the predicted value. The category of programs and rating in each category were not taken into consideration. The performance was measured for different values of T_1, the neighborhood selection parameter. In a case of T_1 = 0.1, mean absolute error turned out to be 1.377, mean square error 3.245 and standard deviation 1.801. The system could not predict 2.885% of programs. The distribution of mean absolute error has a bell shape as presented in Figure 9. The distribution of user's mean absolute error is presented in Figure 10.

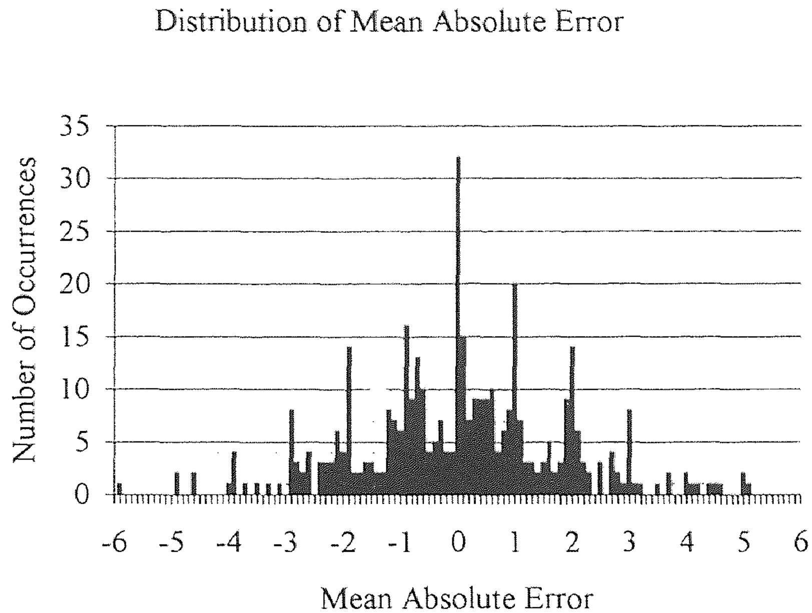

Figure 9 The Distribution of Error in Analysis without Adjusting Factor

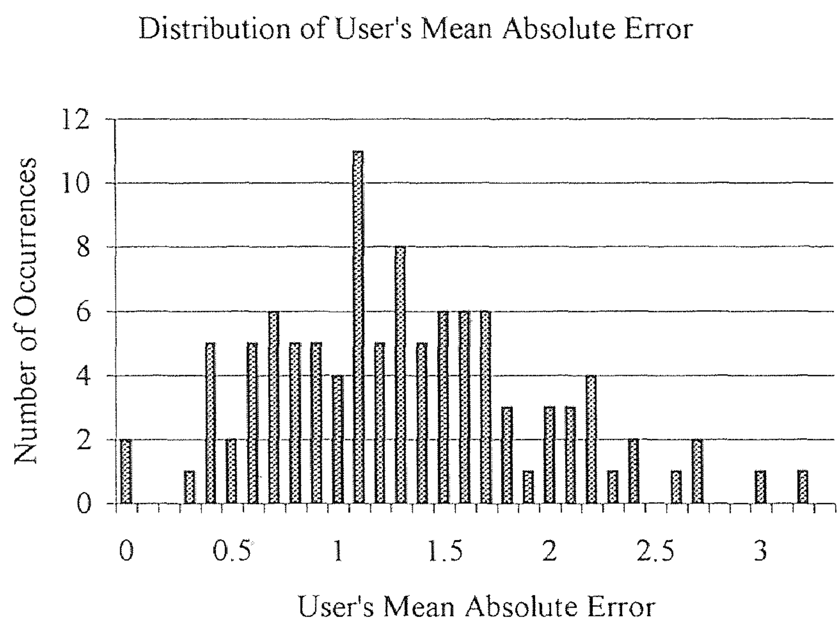

Figure 10 The Distribution of User's Mean Error in Analysis without Adjusting Factor

When the value of T_1 was incremented to 0.2, the mean absolute error increased to 1.364, mean square error to 3.081 and standard deviation decreased to 1.750. The percentage of programs that could not be predicted increased to 18.029%. Further incrementing T_1 to 0.3 generated mean absolute error of 1.399, mean square error of 3.357 and standard deviation of 1.831. The system could not predict 56.989% of program scores. The similar scenario appeared here as also noticed in the analysis of the music recommending system Ringo where, after a certain threshold value the prediction effectiveness started to decline[7]. The reasons are not determined here.

Further, the system was analyzed for a neighborhood selector threshold T_1 of 0.0, where now the neighborhood consisted of all users. Surprisingly, the mean absolute error declined to 1.344, mean square error declined to 3.032 and standard deviation to 1.741. The percentage of programs that could not be predicted declined to 0.481%. From the analysis it appears that the consideration of all users together with the proper weighting of rated programs in the prediction equation may be more effective than a selection of users with similar taste and the generation of recommendation just on that subset of users. A summary of results of analysis without an adjusting factor is presented in Table 1.

Table 1 Results of Analysis without Adjusting Factor

T1	Mean Absolute Error	Standard Deviation	Mean Square Error	Percentage Cannot Predict
0.0	1.344	1.741	3.032	0.481%
0.1	1.377	1.801	3.245	2.885%
0.2	1.364	1.750	3.081	18.029%
0.3	1.399	1.831	3.357	56.989%

4.4 Analysis with Adjusting Factor

The system was altered to include an adjustment of the predicted score based on an average rating of the prediction requester in a given program category. The system was evaluated for different values of neighborhood selecting threshold T_1 and score adjusting threshold T_2. At first T_1 was set to 0.1 and T_2 to 0.1. The mean absolute error turned out to be 1.294, mean square error 2.870, standard deviation 1.693 and the system could not predict 15.625% of programs' scores. The distribution of mean absolute errors is presented in Figure 11 and distribution of user's mean absolute error in Figure 12.

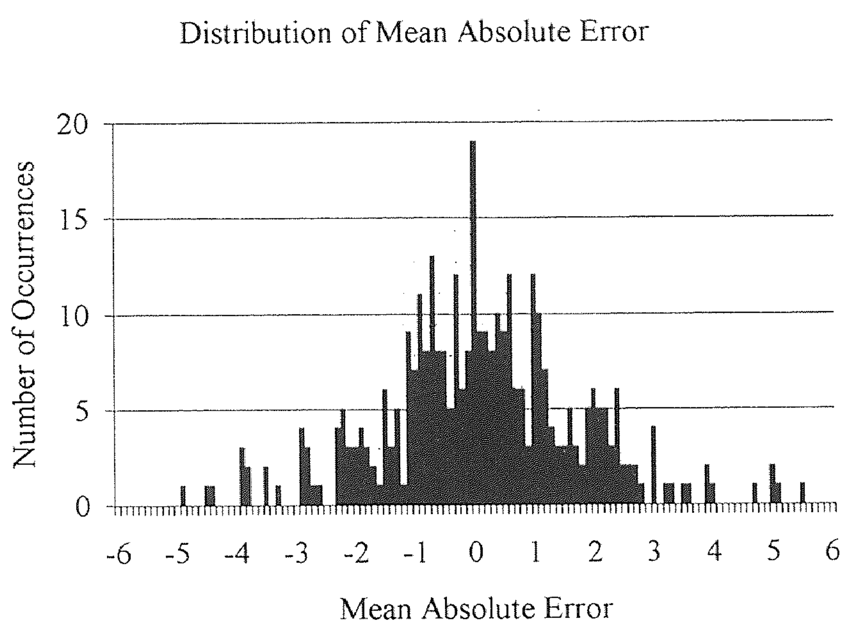

Figure 11 The Distribution of Error in Analysis with Adjusting Factor

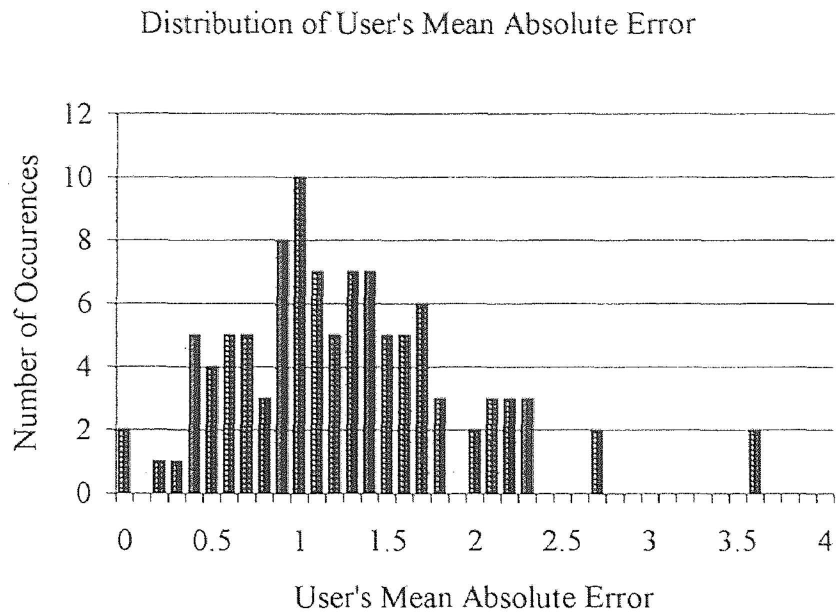

Figure 12 The Distribution of User Mean Error in Analysis with Adjusting Factor

Then, while T_1 was held at 0.1 and the value of T_2 was incremented to 0.2, the mean absolute error increased to 1.300, mean square error increased to 2.913 and the standard deviation to 1.704. The percentage of programs that could not be predicted remained the same. When the value of T_2 was incremented to 0.3, a similar scenario as in prior analysis was noticed where the mean absolute error incremented to 1.353, mean square error to 3.109 and standard deviation to 1.760, leaving the percentage of programs not being able to predict the same.

Then the value of T_1 was incremented to 0.2 and the prediction was run for different values of T_2. When T_2 was set to 0.1 the mean absolute error turned out to be 1.348, mean square error 2.952 and standard deviation 1.718. The system could not predict 27.694% of programs. For T_2 set to 0.0 the mean absolute error increased to 1.351, mean square error increased to 3.025 and the standard deviation to 1.738; the percentage of not predicted programs remained the same. The trend had been noticed that the most effective prediction appears to be for $T_2 = 0.1$ for different values of T_1.

Then while the value of T_2 was kept at 0.1, T_1 was decreased to 0.0. The mean absolute error turned out to be 1.273, mean square error 2.730 and standard deviation 1.648. The percentage of not predicted programs was 14.183%. By decreasing T_2 to 0.0, the mean absolute error increased to 1.302 and standard deviation to 1.680. The percentage of not predicted programs remained the same.

From the analysis of predicting scores it appears that the system with adjustment of predicted score is most effective at $T_1 = 0.0$ and $T_2 = 0.1$. The two parameters are independent of each other and both of them are proportional to the mean absolute error , mea square error and standard deviation. The results of analysis considering adjusting factor are summarized in Table 2.

Table 2 Results of Analysis with Adjusting Factor

T1	T2	Mean Absolute Error	Standard Deviation	Mean Square Error	Percentage Cannot Predict
0.0	0.0	1.302	1.680	2.827	14.183%
0.0	0.1	1.273	1.648	2.730	14.183%
0.0	0.2	1.288	1.674	2.823	14.183%
0.1	0.0	1.325	1.736	3.015	15.625%
0.1	0.05	1.307	1.707	2.916	15.625%
0.1	0.1	1.294	1.693	2.870	15.625%
0.1	0.2	1.300	1.704	2.913	15.626%
0.1	0.3	1.353	1.760	3.109	15.625%
0.2	0.0	1.351	1.738	3.025	27.644%
0.2	0.1	1.348	1.718	2.952	27.644%

4.5 Conclusions

The analysis of TV Recommender established following patterns:

1. ***Considering all users with appropriate weighting of rated programs is more effective then defining neighbors of similar taste.***

In case of adjusting predicted score as well as in a case of not adjusting the score, the mean absolute error and standard deviation is minimal for value of neighborhood selection T_1 of 0.0. Interesting results had been observed while investigating different values of T_1. The mean absolute error as well as standard deviation seems to increase to a certain point, then it seems to slightly decline, giving an impression of creating a smaller neighborhood therefore a better recommendation. However, surprisingly, further increase in T_1 results in an increase of mean absolute error and standard deviation. The reason for that may be the use of the same parameter T_1 for neighborhood selection as well as for weight calculations. By considering only users with similar taste, the recommendation can be made but since the distribution of the correlation coefficient has a bell shape around the value 0, a lot of useful data is not utilized. By appropriately weighting each user in the system based on the correlation coefficient relative to the prediction requester, a more precise prediction and recommendation can be made.

2. ***Adjusting predicted scores increases performance of the recommending system.***

It is evident that program prediction becomes more accurate when it is adjusted based on a prediction requester taste of programs in a specific category, not just opinions of other users. The reason for the increase in the percentage of programs that can not be predicted is the same as the reason for the decrease in the mean absolute value and standard deviation. If the prediction requester did not rate any programs in a given category, the system does not know the taste of the user in that category, and therefore can not predict programs belonging to the category. Also, while users have similar overall TV program tastes, they may differ greatly in a specific category of programs. Without adjustment of the predicted score, a recommendation can be made, but the final scaling of the result can generate a much more precise prediction.

4.6 Quantitative Results

The source of qualitative analysis is a collection of comments from users of the TV Recommender. Many of the users provided some feedback about their TV programs recommendation. The results are as good as source provided by users. It is expected that the analytical data would not necessarily be reflected in verbal responses; however, in the case of TV Recommender, a strong similarity was noted.

The performance of TV Recommender rises with the growth of the system database, specifically the number of people that rated a number of TV programs. Originally, the system was set with an algorithm not taking into consideration the division into categories and not adjusting predicted score. The neighborhood selection threshold was set to 0.4, but very quickly it was changed to 0.1, because otherwise the users would have to wait for any prediction for a long time. Users' responses were very positive. At first, some of them admired the idea of personal recommendation, some gave advice on a friendlier user interface and possible extra features. They were not as critical on the precision of the recommendation, understanding that the system did not have enough data for precise prediction. Later, users were more concentrated on the recommendation itself. There were many responses stating that the recommendations are very good and that they enjoyed the idea of personal recommendation. For most of the cases, however, users predictions for some programs were very precise and for others not as accurate. The most common complaint was that the system did not let users add new programs to the database. There were users stating that their ratings are not as precise because they are viewers of different TV stations and the programs that they would like to rate and obtain recommendation for are not included in the TV Recommender database. Surprisingly, those were the individuals for whom the mean absolute errors were the greatest. The problem probably could be avoided by allowing users to add new programs, what was provided at the end.

There were also a number of users who did not trust that any personal recommendation system could ever determine their TV taste. Some others questioned how the system could determine personal recommendation on a certain programs if the user did not rated any of the similar programs. That was the inspiration to include categories of programs and adjust the predicted score based on average rating for programs in that category.

The design, development and analysis of the TV Recommender had been a very enjoyable experience. The design and development contained a lot of unknowns, for which the solutions could not be speculated until data started to be collected and analyzed. The responses from users were very exciting and informative. It is a very exciting concept of studying human behavior through user's responses to a system such as the TV Recommender. The particular trends could be determined based on a whole population of users and the relationship among them could be defined and analyzed. In many cases the verbal responses from some users could positively confirm the obtained statistical results.

CHAPTER 5

FURTHER RESEARCH

The TV Recommender system was built to collect user data and to determine effectiveness of a prediction algorithm. The analysis showed that the prediction could be determined and presented some ways to improve precision of the recommendation. However, there are still a number of further experiments that can be explored in the future.

- ***Analysis of the algorithm with more data***. Due to the fact that the system existed only for a short period of time, the user data and, thus program data, was limited. There were many users willing to recommend different programs that were not included. Many people may have been forced to rate listed programs, not the programs that they would like to rate. That may have effected the performance of the system. A further analysis can be done when the data in a system becomes more stable.

- ***Analysis of the algorithm in a different domain.*** The further research can be explored in a domain other than TV programs. It would be very interesting to see if the determined patterns would have similar results as those determined for TV programs and also if the threshold values would remain the same or would have to be adjusted accordingly.

- ***Analysis to determine minimum sufficient data.*** The largest concern in analysis of systems similar to the TV Recommender is collection of data and determining at what point there is a sufficient amount of data for analysis. There are many factors influencing the generation of the prediction and collection of data that may produce significant results. It may be questionable at what point the data gathered by the system is sufficient to determine possible trends. An analysis could be done to determine the minimum number of programs in a database, how many programs a user must rate in order to establish some results and to determine other ways to minimize user data collection time.

- ***Automatic threshold adjustment.*** In order to determine the effectiveness of any system similar to the TV Recommender, the process must be run multiple times and appropriate parameters needs to be adjusted. An algorithm could be determined to automatically adjust all threshold values for the recommendation engine to be the most precise.

- ***Analysis of Distribution of Programs in Categories.*** A number of experiments can be done to find out the optimum categories for TV Recommender. An analysis can be done to determine if it is beneficial to allow users to control the programs categories or if it should be system static.

- ***Analysis of Adjustment of Predicted Value.*** The TV Recommender proposed an adjustment of the predicted value based on an average rate of prediction requester for specific categories of programs. Different equations to adjust predicted score could be examined which do not have to be linear like the one used in TV Recommender. One can be to adjust the score based on a difference in taste in specific category between the prediction requester and the people in database ratings based on which the prediction is calculated.

- ***Enhancements of Recommendation Engine.*** The prediction building procedure still could be examined to add more factors that influence the effectiveness of prediction. It would be interesting to examine if users favor, for example, specific TV stations, and include that as part of the recommendation engine. Content based filtering could also be incorporated. The recommendation engine could also consider the number of people that rated specific program and adjust the prediction for that program appropriately. Other ways of finding the relationship between the prediction requester and all other users can be analyzed.

- ***Analysis of User Interface.*** In the generation of predictions, there is the factor of human inconsistency of rating programs. The same person may rate the same program slightly different at different times. Experiments could be done with the user interface to determine the optimal presentation of programs to rate in order to minimize the error. One could try to present users with the average rating for randomly chosen programs and ask users to adjust the rate accordingly, rather than present users with no ratings. An analysis could be done to determine and minimize the human inconsistency factor in the generation of recommendations.

- *Avoid Abuse.* Last but not least, a number of possible violations that could be made by unethical users needs to be determined and processes to detect and stop the corruption of the system needs to be develop. In order for the system to be maintenance free, users must comply with specific rules and regulations that need to be forced upon them. Examples of such unwanted behavior in systems like TV Recommender could be flooding the system with fictitious users or programs, or assigning incorrect categories to added programs.

APPENDIX A

PAGES FROM WORLD WIDE WEB INTERFACE

Welcome to TV Recommender

(New Features: you can add programs and view your profile now!!!)

Program 1 Graphics	**Program 2 Graphics**	**Program 3 Graphics**

New User | Login

Program 4 Graphics	**Program 5 Graphics**	**About TV Recommender** **About Collaborative Filtering** **Help Using TV Recommender**	**Program 6 Graphics**

Created by Elizabeth Podberezniak Advised by Prof. John Carpinelli

Note: The system uses graphics of current TV programs. Due to copyright infringements, for the purpose of the thesis the graphics are removed.

Elizabeth, Please Rate 20 or more TV programs

Remember, more programs you rate, better recommendation you get

Program 1	Not my taste	Program 2	It is okay
Program 3	Dont know	Program 4	It is okay
Program 5	Hate it	Program 6	Great stuff
Program 7	I like it	Program 8	Dont know
Program 9	Dont know	Program 10	Dont know
Program 11	The best	Program 12	Not my taste
Program 13	Dont know	Program 14	Dont know
Program 15	Pretty bad	Program 16	Great stuff
Program 17	The best	Program 18	Dont know
Program 19	Dont know	Program 20	Dont know
Program 21	Dont know	Program 22	Dont know
Program 23	Dont know	Program 24	Dont know
Program 25	Dont know	Program 26	Dont know
Program 27	Dont know	Program 28	Dont know
Program 29	Dont know	Program 30	Dont know

Submit

Note: The system uses current TV programs. Due to copyright infringements for the purpose of the thesis, the names are replaced with generic names.

Elizabeth, Here are Your Options

Your Personalized Choices:

- List Programs that You'll Like
- List Programs that You'll Hate
- Give me the Recommendation Rate for ----Choose a Program---- Give
- Change my rate for ----Choose a Program---- as ----Choose a Rate---- in my profile. Add Rate
- Display My Profile NEW

General Choices:

- List the ----Choose a Category---- Programs List
- Add a program (type in the name of the program) which belongs to ----Choose a Category---- category. Add Program NEW
- View comments about ----Choose a Program---- View
- Add a comment about ----Choose a Program----

Comment: Add Comment Reset

[Home] [About] [Help] [FAQ] [Top] [Feedback]

List of Programs You May Enjoy and Recommendation Degree

☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺

Program 1	5.8
Program 2	5.4
Program 3	5.1
Program 4	5.0
Program 5	5.0
Program 6	4.9
Program 7	4.8
Program 8	4.8
Program 9	4.7
Program 10	4.6
Program 11	4.6
Program 12	4.5
Program 13	4.5
Program 14	4.5
Program 15	4.5
Program 16	4.4
Program 17	4.3
Program 18	4.3
Program 19	4.2
Program 20	4.2

Legend for recommendation degree:

1: You'll Hate It 2: Pretty Bad 3: Not Your Taste 4:It's OK
5: You'll Like It 6: Great Stuff 7: The best

How close is your recommendation? Send me your feedback.

[Home] [About] [Help] [FAQ] [Top] [Feedback]

Note: The system uses current TV programs. Due to copyright infringements, for the purpose of the thesis the names are replaced with generic names.

Like of Programs You Would Hate and Recommendation Degree

Program 1	1.0
Program 2	1.9
Program 3	2.0
Program 4	2.3
Program 5	2.3
Program 6	2.4
Program 7	2.4
Program 8	2.4
Program 9	2.4
Program 10	2.5
Program 11	2.5
Program 12	2.5
Program 13	2.5
Program 14	2.6
Program 15	2.7
Program 16	2.7
Program 17	2.8
Program 18	2.9
Program 19	2.9
Program 20	2.9

Legend for recommendation degree:

```
1: You'll Hate It  2: Pretty Bad 3: Not Your Taste   4:It's OK
5: You'll Like It  6: Great Stuff 7: The best
```

How close is your recommendation? Send me your feedback.

[Home] [About] [Help] [FAQ] [Top] [Feedback]

Note: The system uses current TV programs. Due to copyright infringements, for the purpose of the thesis the names are replaced with generic names.

Single Program Recommendation Degree

Your rate for Program 1 is 5.0

Legend for recommendation degree:

1: You'll Hate It 2: Pretty Bad 3: Not Your Taste 4:It's OK
5: You'll Like It 6: Great Stuff 7: The best

Do you agree? Send me your feedback

[Home] [About] [Help] [FAQ] [Top] [Feedback]

Note: The system uses current TV programs. Due to copyright infringements, for the purpose of the thesis the names are replaced with generic names.

Most Liked Programs

Item	Program	Average Rate
1.	Program 1	5.6
2.	Program 2	5.5
3.	Program 3	5.3
4.	Program 4	5.3
5.	Program 5	5.2
6.	Program 6	5.2
7.	Program 7	5.2
8.	Program 8	5.1
9.	Program 9	5.0
10.	Program 10	4.9
11.	Program 11	4.8
12.	Program 12	4.8
13.	Program 13	4.8
14.	Program 14	4.8
15.	Program 15	4.8
16.	Program 16	4.7
17.	Program 17	4.7
18.	Program 18	4.7
19.	Program 19	4.7
20.	Program 20	4.6

Legend for recommendation degree:

1: You'll Hate It 2: Pretty Bad 3: Not Your Taste 4:It's OK
5: You'll Like It 6: Great Stuff 7: The best

[Home] [About] [Help] [FAQ] [Top] [Feedback]

Note: The system uses current TV programs. Due to copyright infringements, for the purpose of the thesis the names are replaced with generic names.

Most Popular Programs

Item	Program	Number of Ratings
1.	Program 1	62
2.	Program 2	61
3.	Program 3	57
4.	Program 4	56
5.	Program 5	55
6.	Program 6	54
7.	Program 7	54
8.	Program 8	53
9.	Program 9	52
10.	Program 10	52
11.	Program 11	51
12.	Program 12	51
13.	Program 13	49
14.	Program 14	49
15.	Program 15	49
16.	Program 16	48
17.	Program 17	48
18.	Program 18	46
19.	Program 19	46
20.	Program 20	45

[Home] [About] [Help] [FAQ] [Top] [Feedback]

Note: The system uses current TV programs. Due to copyright infringements, for the purpose of the thesis the names are replaced with generic names.

APPENDIX B

HELP SCREENS

About TV Recommender

Welcome to TV Recommender.

We all spend a lot of time watching TV but really we spend a lot of time just finding what is on and deciding if we would like it. Here is a perfect tool. The TV Recommender will be able to tell you which programs you should check out and which programs you should avoid. All you have to do is provide the system with a sample of your TV program taste and the Recommender will create your own personal recommendation of TV programs that you would like and the ones that you would not like as much. Along with the recommendation there are also some features to list the most liked and the most popular TV programs, rate individual programs and more.

So, be my guest, check it out.

The TV Recommender is a research project for New Jersey Institute of Technology in collaborative filtering.

[Home]

Frequently Ask Questions

My recommendation is not available

The reason may be that there are still not enough people registered. Remember that your recommendation is derived from recommendation of others, so bother other people to register and increment the population. The more people will register, the more recommendations the system can come up with.

If you are asking for a specific program recommendation then most probably you have rated that program yourself.

Another reason may be that you rated too many programs. If you rated all the programs the system does not have any left to recommend. If that happened, register with a different name and start again.

My recommendations are not accurate

You probably did not rate enough programs. Also at the beginning the system does not have enough people and then the recommendation will not be accurate. Bother others to join and build up the system.

Whatever I try to add or list, I get Failure message

Make sure that you choose whatever is necessary from the drop box(es) for the item requested.

Comments and suggestions, please mail to Elizabeth Podberezniak.

[Home] [Top]

Help Using TV Recommender

The system is really simple and does not need too much explanation. Mainly, enter as New User (if that is your first time) or login into the TV recommendation system. The system is self-guided so you should be able to just follow.
You will be presented with 30 randomly chosen TV programs and a pull down menu next to each program. Please rate as many programs as you can, but rate only programs that you have seen and, try to be honest. That is the only way the system will know your preferences. Remember that based on those preferences your recommendation is built on. If you can not rate 20 programs on that page submit what you can and the system will provide anther 30 randomly chosen programs. If there are any of the same programs your rate will also be included.

When you complete rating at least 20 programs you would be provided with a few options:

Your Personalized Choices:

List Programs that You'll Like provides your personalized list of up to 20 TV programs that the TV Recommender thinks that you would enjoy watching excluding programs that you have rated and a recommendation degree the system thinks at which you would like the program. The degree level is included here and it is also displayed on the recommendation screen.

```
7 - The Best
6 - Great Stuff
5 - I like It
4 - It's OK
```

List Programs that You'll Hate provides your personalized list of up to 20 TV programs that the TV Recommender thinks that you would not enjoy and should avoid watching. The recommendation also include the recommendation degree and its explanation which is as follows:

```
1 - Hate It
2 - Pretty Bad
3 - Not My Taste
4 - It's OK
```

Give Recommendation Rate choose this one if you would like to know how much you would like a particular program. Just choose a program and the system will come back with a suggested recommendation level:

```
0 - Don't Know
1 - Hate It
2 - Pretty Bad
3 - Not My Taste
4 - It's OK
5 - I Like It
6 - Great Stuff
7 - The Best
```

Change Rating for a Program changes rating of a particular program to your profile. By having a larger profile the system will know you better and therefore the recommendation will be more accurate.

Display My Profile displays a content of your profile: lists programs that user rated and score that you gave to each program

General Choices:

List Programs lists programs based on the category chosen:

1. 20 Most Liked programs based on an average rate and their average rate

2. 20 Most Popular programs based on a number of participants that rated that program

3. All programs and their average rate as well as their popularity number

Add Program to Database adds a specified program of the specified category into the TV Recommender database and makes it available for ratings

View Comments	view a list of comments about a specific program that other users have left to share
Add Comment for a Program	adds a comment about a specific program for other users to view; you may or may not sign your comment

Comments and suggestions, please mail to Elizabeth Podberezniak

[Home] [Top]

APPENDIX C

USER COMMENTS

It works pretty good. It recommended my favorite show as #1. It was right-on with my least favorite. However, it did miss on a couple, notably, since I didn't much like 48 Hours. It figured I didn't much like 20/20, which I like.

Obviously, the current results are pretty skewed by your small user base. The big surprise for me was that I should hate Tom Snyder (who I've always loved) and love Monday Night Football. (I guess you're never too old to learn things about yourself.)

The first recommendation for me I do like; the second I can't stand. Five of those listed are shows I really don't like. It is not a very good prediction.

My anecdote: I think Mad About You is a bit boring, but Frasier is better (still about the same though).

My ratings were pretty accurate. I do love 21 Jump Street and you indicated I wouldn't like it. Also, I don't LOVE the Simpsons but would put them higher than you ranked me. Other than that, I think that you were very accurate. Some of the shows I've never heard of though, and can't comment (i.e. Strange Universe).

Your TV Ratings are not too far off. Good job and good luck! I will include a link to your site on my webpage when it goes online March 1, 1998.

Your rate for All My Children is 1.9. AMC's okay. I watch/tape it at least once a week. Your rate for As the World Turns is 1.3. I don't watch ATWT now, but I used to. Your rate for Business Center is 4.0. There is no way that I would watch it.

I'm a very big fan of the soap genre, daytime & primetime & space soaps, but this shows me as uninterested in the ones you have.

Well you don't have my favorite Soap Opera listed so how can it accurately rate what I want to watch when it doesn't give all possibilities. My Favorite is Another World. DRool and SB are listed but not AW now is that fair? They are all 3 NBC soaps!

I agree with your ratings. The recommendations were very close to the programs that I would watch your list of programs that I would dislike is very good. There are only two programs listed that I like to watch: The Peoples Court, and Grace Under Fire. The recommendation for MASH is correct.

The recommendations her program made to me were pretty close to accurate - I don't agree with the order, but the general "You would like these shows" and "You wouldn't like these shows" were more or less accurate.

REFERENCES

1. Firefly Networks Inc. "Collaborative Filtering Technology: An Overview."(1996, 1997). http://www.firefly.net/products/CollaborativeFiltering.html. (5 Sept 1997).

2. D. Maltz and K. Erlich, "Pointing the Way: Active Collaborative Filtering", *Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems* ACM, New York, 1995.

3. D. Goldberg and D. Nichols and M. B. Oki and D. Terry, "Using collaborative filtering to weave an information tapestry", *Communication of the ACM*, vol. 35(12), pp. 61-70, 1992.

4. P. Resnick and N. Iacovou and M. Suchak and P. Bergstrom and J. Reidl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", Center for Coordination Science, MIT Sloan School of Management Report WP#3666-94.

5. "The GroupLens Recommendation System", (1997), http://www.cs.umn.edu/Research/GroupLens/clientLib/clientGuide.html, (27 Jan. 1998).

6. Firefly Networks, Inc. "Building Intelligent Relationships: The Firefly Tools", (1996-1997), http://www.firefly.net/products/FireflyTools.html, (5 Sept. 1997).

7. U. Shardanand "Social Information Filtering for Music Recommender", MIT EECS M. Eng. Thesis, also TR-94-04, Learning and Common Sense Group, MIT Media Library, 1994.

8. U. Shardand and P. Meas "Social information filtering: Algorithms of automating "word of mouth"". *Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems* ACM, New York. pp 210-217. 1995.