Universidade Federal de Pernambuco
Centro de Informática

Group Recommendation Strategies Based on Collaborative Filtering

Sérgio Ricardo de Melo Queiroz

Recife
2003

Sérgio Ricardo de Melo Queiroz

GROUP RECOMMENDATION STRATEGIES BASED ON COLLABORATIVE FILTERING

ESTRATÉGIAS DE RECOMENDAÇÃO PARA GRUPOS BASEADAS EM FILTRAGEM COLABORATIVA

Dissertação apresentada à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Francisco de Assis Tenório de Carvalho

Recife, outubro de 2003

Abstract

Nowadays, the amount of information available far exceeds our ability to manage it. We can choose from dozens of TV channels, thousands of movies, millions of books, billions of on-line documents. When we have to make choices without full knowledge of the alternatives, a common approach is to rely on recommendations of trusted persons.

In the 1990s computer recommender systems have appeared to automatize the recommendation process. Today, popular sites like Amazon.com give thousands of recommendations every day. However, while many activities are carried out in groups, like going to the movies with friends, current systems focus only on recommending for sole users. This brings out the need of systems capable of performing recommendations for groups of people, a domain that has received little attention in the literature.

In this work, we investigate the problem of generating automatic group recommendations, making connections with problems considered in other research areas like social choice and social psychology. We propose two methods based on collaborative filtering to generate recommendations: one that aggregates individual recommendations based on an existing technique of classification of alternatives which uses fuzzy majority; and a novel methodology that builds a model for the group using techniques from symbolic data analysis. Finally, we empirically evaluate the proposed methods to see their behavior for groups of different sizes and degrees of homogeneity. To this end, we develop an evaluation framework that quantifies the quality of the group recommendations based on a set of metrics that reflect desirable properties these recommendations should have.

Keywords: recommendations for groups, recommender systems, collaborative filtering, symbolic data analysis, information filtering.

Resumo

Atualmente, a quantidade de informação disponível é muito maior do que nossa capacidade de tratá-la. Podemos escolher entre dezenas de canais de TV, milhares de filmes, milhões de livros, bilhões de documentos *on-line*. Quando temos que fazer escolhas sem conhecimento completo das alternativas, uma saída comum é recorrer a recomendações de pessoas de confiança.

Na década de 1990 surgiram os sistemas de recomendação computacionais, que automatizam o processo de recomendação. Hoje em dia, *sites* populares como Amazon.com fornecem milhares de recomendações todos os dias. No entanto, enquanto muitas atividades são realizadas em grupos, como ir ao cinema com amigos, os sistemas atuais dedicam-se apenas a recomendações para usuários individuais. Isto desperta a necessidade de sistemas capazes de realizar recomendações para grupos de pessoas, um domínio que tem recebido pouca atenção na literatura.

Neste trabalho, nós investigamos o problema de gerar recomendações automáticas para grupos, fazendo conexões com problemas considerados em outras áreas de pesquisa como escolha social e psicologia. Nós propomos dois métodos baseados em filtragem colaborativa para gerar recomendações: um que agrega recomendações individuais baseado em uma técnica existente de classificação de alternativas que utiliza maioria nebulosa (*fuzzy*); e uma metodologia inédita que constrói um modelo para o grupo usando técnicas de análise de dados simbólicos. Finalmente nós avaliamos empiricamente os métodos propostos para verificar o seu comportamento em grupos de diferentes tamanhos e graus de homogeneidade. Para tal nós desenvolvemos um modelo de avaliação que quantifica a qualidade das recomendações para grupos baseando-se em um conjunto de propriedades desejáveis para estas recomendações.

Palavras-chave: recomendação para grupos, sistemas de recomendação, filtragem colaborativa, análise de dados simbólicos, filtragem de informação.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The new millennium is the information age. In the 1990s, there was an explosion in the amount of available information. People can choose from dozens of TV channels, thousands of movies, millions of CDs and books, billions of on-line documents. Nonetheless, our ability to manage information remains the same. Therefore, people find themselves shipwrecked in the middle of an ocean of information.

What can be done? When one has to make choice without full knowledge of the alternatives, a common approach is to rely on the recommendations of trusted individuals: a movie critic, a friend, or a consulting agency.

This scenario allowed the flourish of computational recommender systems. These systems automatize the recommendation process.

Nowadays, we have (mostly in the Web) various recommender systems. Popular sites like Amazon.com have recommendation areas where the individual can see which items would be of his/her interest. Every day, these systems give thousands of personalized recommendations. However, until now, these systems have focused only on making recommendations for individuals, despite the fact that many day-to-day activities are performed in groups, such as:

- Watch TV at home.
- Go to the movies with friends.
- Listen to the radio in the car during a family trip.

Consequently, if one wants to go to a movie theater with his/her friends, a recommendation, to be useful, has to be adequate for the group as a whole, and not only for one individual.

That points to the need of developing recommender systems for groups, that are capable of capturing the preferences of whole groups and make recommendations for them.

1.2 Goals

In this dissertation the problem of making recommendations for groups is analyzed. Two different methods of making recommendations for groups are proposed, and after they are empirically analyzed. In summary, the main goals of this dissertation are:

- Pose the problem of making recommendations for groups, pointing out the inherent difficulties of the problem.
- Analyze methodologies for treating the problem. The methodologies used are based on the principles of collaborative filtering, one of the most successful methods for making recommendations (for individuals).
- Analyze the behavior of the presented methodologies. For this, real data is used to empirically observe the behavior of the presented methodologies for groups of different characteristics.

To better characterize the problem of recommendation for groups, results from related research fields are considered. These results show that a perfect recommendation strategy for groups is an unachievable goal. We propose two different strategies to make recommendations for groups, and after setting a framework for empirically evaluating them, we analyze them under varying group sizes and different levels of agreement for the preferences of the group members.

1.3 Organization of the dissertation

An outline of the remainder of the dissertation can be seen in Table 1.1.

Table 1.1 Overview of the dissertation.

Chapter	*Chapter Description*
2	**The recommendation problem**. This chapter describes the recommendation problem and introduces collaborative filtering, the most used method used to generate recommendations in the field of recommender systems.
3	**Related Work**. Describes previous research in the recommender system literature as well as other fields related to the problem of this dissertation.
4	**Recommending for groups using aggregation-based methodologies**. Presents one approach for making recommendations for groups: by first recommending for individuals and them aggregating the recommendations. An existing method for the classification of alternatives using fuzzy majority is used to aggregate the users' recommendations.
5	**Recommending for groups using model-based methodologies**. In this chapter, a novel methodology for generating group recommendations is developed. It first builds a model for the group that wants the recommendation and them generates the recommendations directly for this model.
6	**Experimental Design and Evaluation Metrics**. Here we develop the framework we use to empirically evaluate the different recommendation methodologies.
7	**Results and Discussion**. The behavior of the recommendation methodologies under the evaluation framework proposed in Chapter 6 is analyzed.
8	**Conclusions**.

Chapter 2

The recommendation problem

In this chapter, the problem of recommendation is presented. We also present collaborative filtering, the most successful technique used to make recommendations (for individuals).

2.1 Overview

The use of recommendations is commonplace in people's routine. It is common to read movie reviews to help decide which movie to see. Or ask a bookseller to suggest a book your science-fiction fanatic acquaintance will probably like. The perception of a restaurant which is always full indicates that it is probably a good place to eat, so you decide to give it a try.

These examples help understand the concept of a recommendation. In general, one individual is facing a decision, a choice given an universe of alternatives. This universe is typically enormous, even making it impossible to the individual to know which are all the alternatives available. Therefore the task to choose between these alternatives is extremely arduous [63].

The usefulness of recommendations is not limited to recommendations focused on lone individuals. Many activities are carried out in groups (e.g. watching TV at home, going to the movies with friends, listening to the radio in the car during a trip with the family). Even some traditionally solitary activities (like web browsing) are sometimes performed in groups. WebTV has estimated that two people were present on average during a browsing session in its service, indicating that in this case browsing in groups was the norm rather than the exception [34].

To treat this problem of *information overloading*, different techniques exist to find the informations needed by the user (*filtering in*) and to eliminate the unneeded ones (*filtering out*). The term *information filtering* is used to refer to these two acts. Malone et al. [37] identified three categories of information filtering:

- **Cognitive filtering**: selects the information based on its contents. The mail filter rule "send all messages that contains the string 'make money' directly to trash" is an example of cognitive filtering.
- **Social filtering**: it is based on the relation between people and their subjective judgments. The mail filter rule "send all messages from Jeff K. directly to trash" is a simple example of social filtering.
- **Economic filtering**: is based on the relation cost/benefit of producing an item. An economic mail filter could use the rule: "if a message has been sent to many recipients it has a small production cost by address, therefore it should have low priority; on the other hand, a message that has been sent exclusively to the address of the recipient has a high production cost, therefore it should receive higher priority".

The 1990s watched the flourishing of computational recommender systems that automatize the process of recommendation by using information filtering techniques.

The computational recommender systems (from now on referred simply as *recommender systems*) are based primely on techniques from two categories of information filtering: the cognitive filtering, also known as content-based filtering; and the social filtering.

The content-based systems use only the historical preferences of the user to make recommendations for him/her; they try to recommend items similar to what the user has liked in the past (e.g. [36]). The focus of these systems is to learn the user's preferences and find among the items (unknown by the user) those that are the most similar to these learned preferences. Section 2.2 addresses content-based filtering.

The systems based on social filtering use the *collaborative filtering* technique. The focus of this technique is to find users with tastes similar to the user that wants a recommendation (referred as the *active user*), and then recommend items that these "neighbors" have liked. Collaborative filtering is currently amply used, in various domains. For example, the e-commerce site Amazon.com[1] uses collaborative filtering to recommend items to buy, and the music service MusicMatch[2] uses it to recommend songs. Section 2.3 addresses collaborative filtering.

Using one (or a combination of the two) techniques, recommender systems have been able to tackle the problem of recommendations for individuals. However, the problem of recommendations for groups has been mostly ignored. Section 2.4 points out the problems that arise when we need to consider groups, and not only individuals anymore.

2.2 Content-based recommendations

Content-based recommender systems find items similar to the ones the individual has liked in the past. The user's preferences are learned by the feedback given by him. This feedback can be *explicit* (for example, the user can give a score to an item) or *implicit* (for example, the amount of time dedicated to read a web page can be used to measure the user's level of interest on it) [43]. From the feedback received and the description of the items, the system is able to create a profile that reflects the user's interests in types of contents. The manner to represent the user's profile depends on the techniques of machine learning and information retrieval used. For example, it is common to use a prototype made of a vector of words with associated weights (see e.g. [14], [32], [4]); or store the descriptions of the items in case-based reasoning (CBR) systems [10].

Content-based recommendation techniques are out of this work. We will focus on collaborative techniques.

2.3 Collaborative filtering

Notwithstanding the successful application of content-based filtering in many domains, this technique has a series of limitations [57]:

- The content of the items must be manipulable by the computer (for example, textual content), or one must manually register attributes for the items. With current technology, it is very difficult to analyze media like sound and video automatically to extract attributes. Many times it is impossible to define attributes manually due to limitations of resources.

1 http://www.amazon.com

2 http://www.musicmatch.com

- Content-based techniques are unable to find items that would interest the individual but are not similar (in terms of content) to other items that the individual had seen before. Therefore, only items similar to the ones known by the user are found.
- Content-based techniques cannot evaluate the content based on subjective dimensions, like quality. For example, it is not possible to differentiate between two texts with very similar content, but with distinct quality: one is well written, whereas the other is not.

The collaborative filtering (CF) technique is based on the fact that the best recommendations for an individual are those given by people with preferences similar to his/her preferences. The process of CF can be generalized in three steps (Figure 2.1):

- **Representation of the input data**: the user express his/her preferences by evaluating items in the system. This evaluations (positives and negatives) reveal the user's interests in specific items, and are stored as the user profile. The simplest manner to store the profile is as a matrix of m items $\times$ n users, where the cells contain the evaluations. In order to have better scalability and/or precision, a low-dimensional representation may be used instead (see [6], [53]). Notice that the evaluations can also be taken implicitly, for example an e-commerce site can consider that an user likes one item when s/he buys it.
- **Neighborhood formation**: to make a recommendation, the system compares the profile of the active user with the profile of other users to find the similarity between them (the metric used to find these "neighbors" can vary). This set of neighbors formalizes the concept of people with similar preferences.
- **Recommendation generation**: finally, using the information derived from the neighbors, the system recommends items to the user, that is the items most liked by the neighbors will be recommended. Again, the mechanics used to generate the recommendation varies with the CF method used.

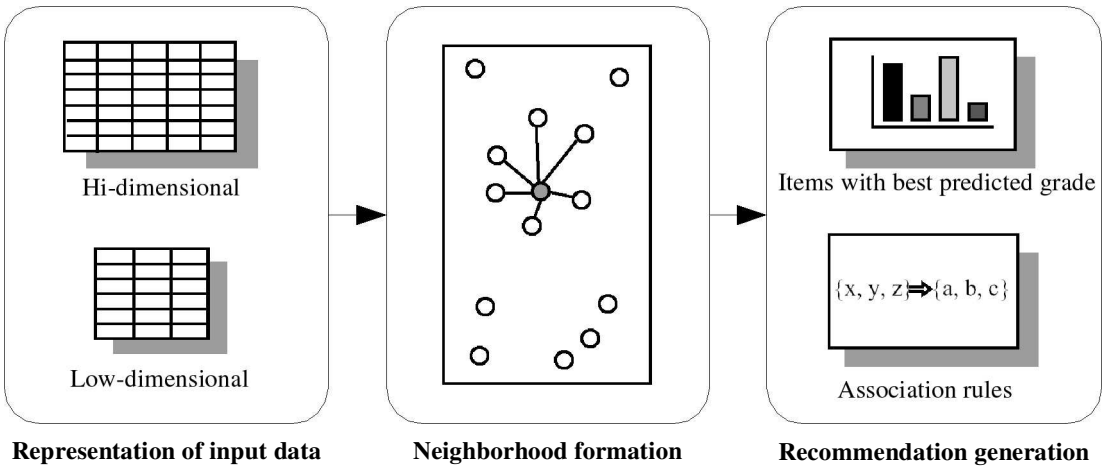


Figure 2.1 Main steps of the CF process (figure adapted from [55]).

To ilustrate the process of collaborative filtering, we will see how it was done in GroupLens [48], a classic recommender system that uses CF based on correlation between users. Even though GroupLens was one of the first CF systems, its CF methodology was later found very effective by Herlocker et al. [20].

2.3.1 GroupLens: a case study

GroupLens is a collaborative filtering system for the Usenet (newsgroups on the Internet). Its goal is to predict how much each article in a newsgroup would interest the user.

When using a news reader compatible with GroupLens, the user (identified by a pseudonym) can evaluate the articles s/he reads. The grades range from 1 to 5, where 1 is the worst and 5 the best grade.

Figure 2.2 shows a matrix of evaluations in an example from [48]. In it, the system contains evaluations given to 6 messages by the users Ken, Lee, Meg and Nan. An empty cell means that the user has not evaluated the corresponding article. Predict how much an article will interest an user means predict the grade this user would give to this unseen article (the cell marked with a "?" will have its grade predicted in the example below).

Article Id	*Ken*	*Lee*	*Meg*	*Nan*
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6	?	2	5	

Figure 2.2 Example of a matrix of evaluations in GroupLens

To make a prediction, GroupLens weighted the users by Pearson correlations and then performed a weighted average from the neighbors' grades. The correlation was computed between the user x for which the grade will be predicted and each one of the neighbors (y) in the system that evaluated the considered item. The Pearson correlation between two users is given by:

$$\rho_{xy} = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y} = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2} \sqrt{\sum_i (y_i - \overline{y})^2}} \tag{2.1}$$

In the above equation, $\overline{x}$ is the mean of the grades of user x. All means and summations in the formula are calculated only for the items that both users evaluated.

Let's suppose that we wish to predict the grade that Ken would give to article 6 (the cell marked with a "?" in the Figure 2.2). The first step is to calculate the correlation between Ken and each other user. The correlation between Ken and Lee according to Equation 2.1 is:

$$\overline{K} = \frac{1+5+2+4}{4} = 3 \quad ; \quad \overline{L} = \frac{4+2+5+1}{4} = 3$$

$$\rho_{KL} = \frac{(1-3)(4-3)+(5-3)(2-3)+(2-3)(5-3)+(4-3)(1-3)}{\sqrt{(1-3)^2+(5-3)^2+(2-3)^2+(4-3)^2}\sqrt{(4-3)^2+(2-3)^2+(5-3)^2+(1-3)^2}}$$

$$= \frac{-2-2-2-2}{\sqrt{10}\sqrt{10}} = -0.8$$

Similarly, the correlation coefficient of Ken with Meg is +1 and with Nan is 0. That is, Ken normally disagree with Lee (ρ_{KL} = -0.8) and agree with Meg (ρ_{KM} = +1). His evaluations are not correlated with Nan's.

To predict the grade of *x* for the article *i*, we take a weighted average of all evaluations received by article *i* according to the formula:

$$x_{i_{Pred}} = \overline{x} + \frac{\sum_{y \in evaluators} (y_i - \overline{y})\rho_{xy}}{\sum_{y} |\rho_{xy}|} \tag{2.2}$$

Therefore, the grade predicted for Ken for article 6 is:

$$K_{6_{Pred}} = 3 + \frac{2\rho_{KM} - \rho_{KL}}{|\rho_{KM}| + |\rho_{KL}|} = 3 + \frac{2-(-0.8)}{|1|+|-0.8|} = 4.56$$

This is a reasonable prediction, for as we can see in Figure 2.2 article 6 received a high grade from someone who normally agrees with Ken and a low grade from someone who normally disagrees with him.

2.3.2 Deficiencies of collaborative filtering

Despite the success of CF in recommender systems, this technique has some limitations, of which the most noteworthy are:

- **Recommendation of new items**: before an item has been evaluated by a minimum number of individuals it is not possible to recommend it, as the system will not have enough information to predict how much any given user would like it.
- **"Black sheep" user**: if the individual who searches for recommendations does not have enough "neighbors" (i.e. almost nobody in the system has preferences similar to his), the system will inevitably shows a low performance, as the recommendations will be based on users that are considerably different from him.

- **Insufficient number of users**: to have a good performance, a CF system needs a large community of users, or there will not be enough neighbors for each user. When the number of items is enormous (like an on-line bookshop), the need of many users is still stronger because the matrix of evaluations will probably be too sparse.

With the goal of overcoming these limitations, various systems and researches have adopted an hybrid approach, combining CF with content-based filtering (see e.g. [4], [52], [58], [18], [45], [59]), as the latter does not have the underlined deficiencies of CF. In fact, CF is very complementary to content-based filtering, as shows Table 2.1. In this way, an hybrid strategy can profit from the best of each technique.

Table 2.1 Content-based filtering versus collaborative filtering

Feature	*Content-based filtering*	*Collaborative filtering*
Recommendation of new items	No difficulties, the content of the item is used to identify if the user would like it.	Cannot be recommended while they were not evaluated by a sufficient number of users.
"Black sheep" user	No difficulties, the recommendation is based only on the preferences of the own user.	Low performance, because it will be impossible to find similar enough neighbors in order to generate high-quality recommendations.
Small number of users	Independent of the number of users.	Low performance, difficult to find adequate neighbors.
Content not interpretable by the computer (ex.: multimedia)	Manual fill of the attributes necessary. This can make the implementation of a recommender system unfeasible .	No difficulties, the recommendation is completely based on the people and their interrelationships. The content of the items does not need to be known.
Evaluation in subjective dimensions	Difficult to implement.	Intrinsic to people's judgments.
Recommendation of serendipitous items	Normally does not happen. The recommended items are similar to the ones the user had seen in the past.	Easy to happen. The items well-evaluated by neighbors can have content distinct to what the user already know.

Although CF methods based on the identification of neighborhoods using statistical correlation are the most used and have shown great accuracy, deficiencies were found in such methods [55]:

- **Reduced covering**: commercial recommender systems are used to recommend from an enormous set of products (for example, the set of books at Amazon.com). In these systems, even the heaviest users have evaluated less than 1% of the available products (1% of 2 million books is 20,000 books). Given the sparsity of the evaluations, the recommender system can be unable to find recommendations for the users, or their quality can be low. One feature that can ameliorate the problem of sparsity is the transitivity of neighbors. For example, in the traditional correlation-based CF, if user John correlates highly with Clara, and Clara correlates highly with Paul, not necessarily John will have a significant correlation with Paul, as they could have evaluated few items in common.
- **Scalability**: to find neighborhoods, a number of operations proportional to the number of users and items is needed. With millions of users and items, a typical recommender system will suffer from serious scalability problems.
- **Synonyms**: in a real scenario, different product names could refer to similar objects. The correlation-based methods are not able to identify such associations, and consider each product differently. For example if consumer *A* buys 2 bottles of soybean oil and consumer *B* buys 2 bottles of corn oil, a traditional recommender system is unable to find the association between this items as "vegetable oil".

This problems point to some limitations of the CF methods based on correlation. To try to overcome these problems, new CF methods have been studied.

Billsus and Pazzani [6] observed that the prediction problem can be transformed into a classification problem, a well-known task in the machine learning field. Using a dimensionality reduction technique to explore the "latent structure" in the matrix of evaluations, they reduced the need of having many items evaluated in common between users. They built a classifier using neural networks and obtained a better performance than unoptimized correlation-based methods.

Aggarwal et al. [1] developed a new graph-based collaborative filtering technique that showed a significantly better performance in the presence of sparse data.

Lin, Alvarez and Ruiz [35] proposed a collaborative filtering technique based on the use of association rules from the data mining arena [2]. According to the authors this technique is able to identify associations not visible to correlation techniques. Experimental results showed a better performance than unoptimized correlation-based methods, however it was not possible to conclude if the performance was better than the one achieved by the method of Billsus and Pazzani. By using data mining methods, that are prepared to treat large volumes of data, this technique is also easier to scale.

Scalability is a major preoccupation in recommender systems. On-line recommender systems can quickly achieve millions of users and they must generate recom-

mendations within few seconds. To facilitate the fulfillment of these requirements, Sarwar et al. [54] developed a method based on the "neighborhood" of items, instead of users. As in a typical recommender system the set of items is much more stable than the set of users, these neighborhoods can be stored off-line, facilitating the scalability of the system. They also found that this method demonstrated better accuracy than methods based on the correlation of users.

2.4 From individual recommendations to group recommendations

Using the techniques cited in Sections 2.2 and 2.3, it is possible to make recommendations for individuals. In fact, as it was previously mentioned, a good number of recommender systems have been successfully deployed.

Until now, these recommender systems have focused on the problem of generating recommendations for individuals. However, many common activities are carried out in groups, such as:

- Watch TV at home;
- Go to the movies with friends;
- Listen to the radio in a car during a trip with the family.

These examples show that the horizon of recommender systems can be enlarged to include groups. In the TV scenario, for example, the advent of interactive TV makes possible the transmission of content focused on the particular spectators, unlike the current broadcast process. In this scenario, content personalization will be a key component [61].

Other scenarios where recommender systems for groups would be useful could also be though of. For example, one can think of a recommender system that indicates which "family resorts" would be most interesting for a given family spend their vacations. Also, a "enterprise recommender system" could help identify which teams are more appropriate to handle some new projects that the company will develop.

2.4.1 Recommending for groups using collaborative filtering

We can pose the problem of recommendations for groups in the domain of recommender systems in the following way:

> How to suggest (new) items that will be liked by the group as a whole, given that we have a set of historical individual preferences from the members of this group as well as preferences from other individuals (who are not members of the group).

Thinking collaboratively, we want to know how to use the preferences (evaluations over items) of the individuals in the system to predict how one group of individuals (a subset of the community) will like the items available. Therefore we would be able to suggest items that will be valuable for this group.

To be used to recommend for groups, the CF methodology has to be adapted. We can think of two different ways to modify it with this goal. The first is to use CF to recommend to the individual members of the group, and then aggregate the recom-

mendations in order to achieve the recommendation for the group as a whole (we will call this approaches "**aggregation-based methodologies**"). The second is to modify the CF process so that it directly generates a recommendation for the group. This involves the modeling of the group as a single entity, a meta-user (we will call this approaches "**model-based methodologies**"). Chapter 4 discusses one method that uses the first methodology, whereas Chapter 5 discusses the second.

Whatever method we use for making suggestions for groups, the ultimate goal is that these suggestions are the best possible for the group. This leads to two fundamental questions:

- What is the best suggestion for a group?
- How to achieve this suggestion?

These questions are very difficult to answer. In fact, as it will be seen in the next chapter, there is no definitive answer to what is the best suggestion for a group.

Chapter 3

Related Work

In this chapter, we see how the problem of recommendation for groups has been treated in the recommender system literature as well as related work elsewhere.

3.1 Overview

How to achieve good group results from individual preferences is an important topic in many research areas, with different roots. Beginning in the XVIII century motivated by the problem of voting, to modern research areas like operational research [23], social choice [3], multicriteria decision making [5] and social psychology [60], this topic has been treated by diverse research communities.

These approaches differ in how and what to focus (for example, empirical versus analytical emphasis, consensus versus choice) and, most importantly, the type of inputs used to find the collective choice (for example, preference orderings, intensity of preferences, justifications, argumentations etc.) [27].

This chapter presents an overview of works that are related in general to the topic covered by this dissertation. They are grouped into two sections: the first—Approaches to the group decision problem—presents the different views of the previously cited areas to the group decision problem; the second—Practical implementations—describes the work on recommender systems for groups we could find in the literature.

3.2 Approaches to the group decision problem

3.2.1 Voting theory

Motivated by problems observed in voting methodologies, mathematicians have been studying for centuries the problem of how to reach a fair group decision from individual preferences. But what can be more trivial than an election? Does not it suffice to count the number of votes that each candidate has received to know the winner(s)? What wrong can be something as elementary as that?

In reality, during all this time, mathematicians have shown that when there are at least three candidates—a common situation—the winner is not always the one preferred by the voters. As said by Saari in [51], "such bad outcomes may occur not only because some voters continue to vote long after death; bad outcomes can also be caused by hidden mathematical peculiarities".

To point out one problem, known as "the paradox of voting", let's see an example, adapted from [51]. A country resolves to promote a poll to decide which action it should take against the "enemy state of the year" (at this hypothetical time, the population is actually heard about such decisions). In this poll, 1.5 million persons participated and expressed their preferences between "diplomatic negotiations" (N), "embargo" (E) or "war" (W) against the rogue nation. The preferences obtained are shown in Table 3.1.

Table 3.1 Preferences of the population, where "⟩" means "is preferred to"

Number of people (in thousands)	*Preferences*
600	$W \rangle N \rangle E$
500	$E \rangle N \rangle W$
400	$N \rangle E \rangle W$

According to Table 3.1, the result using plurality (where each person votes in his/her favorite action) is $W \rangle E \rangle N$, with results 600:500:400. Apparently, war is the choice of the population.

Before sending the marines, let's see if war is really the preferred option from the population's point of view. If this was true, it is expected that the population prefers war to embargo. However, as can be seen in Table 3.2, people interviewed prefer embargo to war.

Table 3.2 Comparing war to embargo according to the preferences of the population

Number of people	*Preferences*	*War*	*Embargo*
600	$W \rangle N \rangle E$	600	0
500	$E \rangle N \rangle W$	0	500
400	$N \rangle E \rangle W$	0	400
	Total	600	900

In the same way, 900,000 persons prefer diplomatic negotiations to war and 1,000,000 prefer diplomatic negotiations to embargo. This contradicts the result we obtained using plurality, as these comparisons between pairs of alternatives indicate that the real opinion of the population is $N \rangle E \rangle W$, the reverse of the plurality ranking.

In the decade of 1780, the French mathematician, philosopher and politician Marie-Jean-Antoine-Nicolas de Caritat Condorcet argued that the results of elections should be established using comparisons between pairs. The *Condorcet winner* is the one that defeats every other candidate in comparisons between pairs. In the example presented, diplomatic negotiations is the Condorcet winner, whereas war is the Condorcet loser.

The Condorcet winner is normally accepted as the true winner between the candidates. However there still are problems. To illustrate just one difficulty, let's use another example[3]. Let's suppose that a computer science department wants to consult its 15 lecturers who work with Artificial Intelligence to decide which textbook to adopt between the alternatives $\{A, I, M\}$. A natural way to find the Condorcet winner

3 From [51].

is by elimination, where after comparing two alternatives, say $\{A, I\}$, the winner is compared with the remaining option, M. The lecturers' preferences and the comparisons are shown in Table 3.3.

Table 3.3 Lecturers' preferences and comparisons to find the Condorcet winner

Number of people	*Preferences*	A	I	A	M
5	$A \rangle I \rangle M$	5	0	5	0
5	$I \rangle M \rangle A$	0	5	0	5
5	$M \rangle A \rangle I$	5	0	0	5
	Totals	10	5	5	10

As Table 3.3 shows, A wins the initial comparison $\{A, I\}$, but is defeated by M in the following one. In both cases the winner has two thirds of the votes, so it seems clear that the lecturers' preferences are $M \rangle A \rangle I$. However, let's analyze this result. We saw that A beats I and M beats A. We have not compared M (our so far winner) and I (our loser). It seems obvious that M will beat I (as it beats A and A beats I), however, contrary to this belief, I beats M by the same two thirds of the votes. In other words, this example defines a cyclic result: $A \rangle I$, $I \rangle M$, $M \rangle A$. The last candidate considered always wins. There is no Condorcet winner nor loser.

Cycles make it impossible to choose an "optimal candidate", and are one example that shows the difficulty on achieving an optimal ranking for a group.

Many other voting methodologies have been proposed, but none of them works universally, there are always cases where unexpected results appear.

3.2.2 Social choice

Finding a way to aggregate individual choices in order to find the best solution for a group may be seen as a problem of how to find a social maximum from individual desires. This is the central problem of the welfare economics. This problem has been analyzed by a multidisciplinary research field, which combines economics and political science, called *social choice*.

In a social choice groundwork, Arrow [3] identified a set of simple, desirable properties that a social function that gives the collective preference from the individual ones should have. Before presenting these properties, let's see some notation.

Definitions

The relationships between two alternatives may be of preference or indifference. Instead of using two relations, one sole relation is used to indicate "preferred or indifferent". The affirmation "x is preferred or indifferent to y" is symbolized as $x\ R\ y$. The notation R_i is used to represent the ordering relation from the point of view of the individual i over the set of alternatives X, whereas the ordering relation for the society as a whole is represented by R.

R is a connected and transitive relation. Symbolically,

Axiom 1: For every x and y, or $x\ R\ y$ or $y\ R\ x$.

Axiom 2: For every x, y and z, if $x\ R\ y$ and $y\ R\ z$ then $x\ R\ z$.

R is said to be a weak ordering relation. The adjective "weak" means that the ordering does not exclude the possibility of indifference, that is, Axioms 1 and 2 do not forbid that for distinct x and y, $x\ R\ y$ and $y\ R\ x$.

P is the strict preference relation: $x\ P\ y$ is defined as $\neg\ y\ R\ x$.

A social function has as its input a n-tuple of individual preference relations and gives a global preference relation. More formally, we have $f : R_1 \times ... \times R_n \rightarrow R$.

Desirable properties for a social function

Arrow identified the following desirable properties for a social function:

1. **Unrestricted domain**: f has unrestricted domain if, and only if (iff, for short), it is defined for all the Cartesian product (that is, for every possible input—any set of individual preferences).
2. **Independence of irrelevant alternatives**: the social preference relation between any pair of alternatives x and y depends only on the individual preference relations between these two alternatives (i.e., adding or subtracting an alternative z will not change the preference relation regarding x and y).
3. **Pareto condition**: if there are items x, y such that for every individual i, $x\ P_i\ y$, them we will have $x\ R\ y$.
4. **Non-dictatorship**: for every x and y in X (the set of alternatives) there is not an individual i such that $x\ R\ y$ iff $x\ R_i\ y$.

Arrow's impossibility theorem

Arrow demonstrated that it is impossible for a social function to have all the properties aforementioned. In this way, any social decision method will have to abdicate from some of the desirable properties.

Consequently, there is no ideal way to aggregate individual preferences to reach a global result. Every method will have some deficiencies, like the deficiencies with the voting methods mentioned in Section 3.2.1.

3.2.3 Social Psychology

One area of the psychology, named social psychology, has also been studying the problem of group decision making. One approach frequently used in small group research is the theory of *social decision schemes* (SDS). A major preoccupation in this area is to understand how individual characteristics are combined to yield a group result [33].

The theory of SDS is widely used to find group responses from individual preferences. It involves three central considerations: the distribution of the group members'

preferences, the rule that combines these preferences (decision scheme), and the means of testing the adequacy of the decision schemes in predicting a sample of observed group decisions (model testing).

The distribution of preferences

The general SDS model assumes that each group member, and subsequently each group, selects one of n mutually exclusive and exhaustive alternatives. For a group having r individual members, their distribution among the n alternatives can be summarized by (r_1, r_2, ..., r_n), where r_j indicates the number of group members who favor the jth alternative. Note that group members are indistinguishable but response alternatives are distinguishable in this expression. Some extended versions of the SDS permit distinctions between individuals, as well as responses of non-discrete nature (see e.g. [24]).

Decision schemes

A social decision scheme is a rule or procedure that combines (usually in algebraic fashion) the various individual preferences (represented by the group distribution of preferences) into a single group decision. Decision schemes can be constructed to represent a variety of different social processes hypothesized to underlie group decision making.

Model testing

An important concern is the comparison of the various plausible decision schemes through a model testing procedure. The results reached using the proposed decision scheme are compared to the observed (real) group responses. If the two results do not differ significantly, the proposed social decision scheme can be considered as a plausible description of the decision process used by the group.

Decision schemes used in SDS research

Empirical results show that the adequacy of a social decision scheme is dependent on the characteristics of the group members (e.g. willingness to argue, previous knowledge) and the type of problem in question. For example, it was observed a leniency bias in jury decision, which suggests that acquittal is easier to defend than conviction. On the other hand, in problem solving or collective recall, correct options frequently win with only one or two supporters in the group, particularly when correct members are confident of their choice [60]. A partial list of decision schemes that have been used in SDS research is showed below (compiled by [24]):

- **Decision schemes based on central tendency**
  - *Mean*: an obvious way to reach the group preference is to take the mean of the individual preferences. However, this solution may represent a position in which every group member is "abdicating" of his preferences, and nobody in the group is sufficiently satisfied in the end.
  - *Median*: similar to the median, but less sensitive to extreme positions.

- **Decision schemes based on consensus**
  - *Majority*: the alternative chosen is the one which is preferred by at least a majority of the group members. Research in SDS showed a great support for the majority rule. However it was also verified that in mixed-motive groups it can lead to an inferior decision, instead of an integration of interests [41].
  - *Plurality*: when a majority does not exist, the alternative favored by the largest number of people is chosen.
- **Faction-attraction decision schemes**: the group members are attracted to the alternatives supported by a substantial portion of the group (faction). As the size of the faction favorable to an alternative grows, the impact of the faction in the decision process also get higher. A version of this decision scheme that have received some empirical support is that the influence of the faction in the decision process is proportional to the square of its size.
- **Coalition-based decision schemes**
  - *Minimum range majority decision*: the coalition formed by the majority of individuals that have the smallest range of preferences dominates the group decision process. Figure 3.1 shows that a majority that had a small difference of opinions was able to dominate the decision process by forming a coalition in order to indicate alternative "2". The alternative chosen by plurality would be "5", but the advocates of this alternative were not able to make a coalition with individuals of other opinions (the advocates of opinions "0" and "23" were too inflexible).
- **Decision schemes influenced by the distance**
  - *Proportional*: the influence of a member on the final decision is proportional to the proximity between his/her original preferences and the "average" preference of the group (smaller distance means larger influence). It supposes that the group has the tendency of not hearing individuals with preferences too uncommon.
  - *Inverse proportional*: the impact of the individual preferences of a group member on the final decision is inversely proportional to the proximity between his/her individual preferences and the "average" preference of the group (larger distance means larger influence). It supposes that the extreme individuals are the most confident and inflexible, so they will have a greater impact on the group decision.

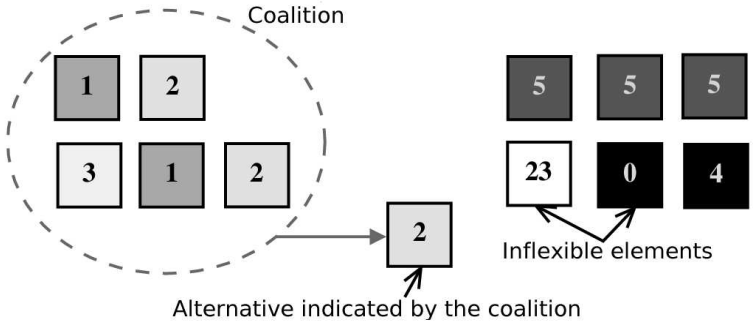


Figure 3.1 Coalition formed by a majority of members with similar preferences dominates the decision process. Here individuals (represented by squares) with opinions "1", "2"and "3" formed a coalition to indicate alternative "2" (that initially had only two supporters), beating alternative "5" which would be the one chosen by plurality. Individuals that supported alternative "5" were unable to make a coalition to indicate their alternative, because the two individuals who supported alternatives "23" and "0" were too inflexible.

- **Dictatorial decision schemes**: some individuals ("dictators") can have a large impact on the final group decision.
  - *Decision scheme of the most able member*: in some types of situation, the presence of a member most able to perform the task is decisive to the group response. For example, in the resolution of puzzle-like problems when one individual discovers the solution, all the group quickly accept it.
  - *Decision scheme of the least able member*: in other types of problems, the presence of a least able member can lead the group to a bad result, impacting the group performance negatively.

3.2.4 Operational research and multicriteria decision making

Operational research is a field born at the 2nd World War to quantitatively analyze different war scenarios in order to indicate which "military operations" would be the most appropriate (therefore the name "operational research"). After the war, this area has focused on business management [23].

Its principle is to search for the best decision, based on the maximization of a "economic function". This paradigm, inspired by physics, has grown deep roots in economy and in many human sciences. In the sixties, the "optimal paradigm" begun to be criticized by specialists in decision aid. In many decision problems, the notion of a "optimum" makes no sense. When you have multiple conflicting criteria, many different results may be pertinent and perfectly legitimate. The "best decision according to all points of view is just a myth" [5].

This is the view taken by multicriteria decision making (MCDM). Hence, MCDM approaches are usually interactive with the goal of aiding the analysis of the decider, not on finding an hypothetical optimum.

We can transform the problem of group decision into a MCDM problem by taking the group as a single collective agent where the preferences of its members are the different criteria under which the problem should be analyzed. In this way, the methods of MCDM can be used to tackle the group problem.

3.2.5 Consequences

Arrow's impossibility theorem show us that there is no perfect method to aggregate individual preferences to reach a group decision. Also, work on social psychology shows that the adequacy of a decision scheme to the group decision process is very dependent to the group's intrinsic characteristics (people's personality, knowledge level, motivations, personal judgments) and the problem's nature (puzzle, analytical problem, jury decision). Multicriteria decision making strengthens the view that the achievement of an "ideal configuration" is not the most important feature when working with decisions (in fact, this ideal may not exist in most of the times) and highlights the importance of giving the users interactivity and the possibility of analyzing different possibilities.

However, the nonexistence of an ideal does not mean that we cannot compare different possibilities. Based on good properties that a preference aggregation scheme should have, we can define meaningful metrics to quantify the goodness of group recommendations. They will not be completely free of value judgments, but these will reflect desirable properties. In Chapter 6 we will look at the problem of evaluating the recommendations.

3.3 Practical implementations

The concept of making recommendations for groups has received little attention in the literature of recommender systems. In this section we cite the efforts in the recommender systems arena to treat the problem of recommendations for groups we could find.

3.3.1 Bellcore video recommender

In one of the first works on recommender systems, Hill et al. [22] stated as one of the design goals of their "virtual community" that recommendations and evaluations should be for sets of people not just individuals. Nevertheless, they did not delve into the difficulties involving the achievement of good recommendation for groups (i.e., the two fundamental questions cited in Section 2.4).

3.3.2 Let's Browse

Let's Browse [34] is a collaborative web browsing agent that uses a content-based approach to recommend web pages for a group of people. A profile that consists of a list of weighted keywords is pre-built automatically for each user, employing a breadth-first search (with constrained depth) starting at the user's homepage. The group profile is a simple linear combination of each user's profile. Pages linked from the current visualized page are recommended if they match the group profile above a

threshold. Therefore, Let's Browse can be seen as a content-based recommender system for groups, with a fixed recommendation strategy.

3.3.3 PolyLens

A more recent initiative, developed by researchers from the GroupLens research group, Polylens [44] is probably the most elaborate initiative to develop a system to recommend for groups in the literature of recommender systems.

It consists of an extension of the MovieLens[4] service for groups of people. MovieLens is a movie recommendation site that generates recommendations for movies based on collaborative filtering technology. MovieLens is used by thousands of individuals, who have provided millions of ratings for movies.

To generate a recommendation for one group, PolyLens first predicts the grade each group member would give for the items (using traditional collaborative filtering). The grade predicted by each item for the group is the smallest one predicted for a member of the group. Therefore PolyLens uses a fixed recommendation policy which assumes that the level of satisfaction of the group is the one of its most miserable member.

Having the live MovieLens site as an experimental framework, PolyLens was able to conduct on-line experiments with the users. This permitted the use of questionnaires to directly ask the users their level of satisfaction with PolyLens and other considerations, like privacy concerns. Also, other practical questions like which interface to use to present group recommendations were investigated. Most users (95%) demonstrated satisfaction with PolyLens and 78% said that group recommendations were more helpful than individual recommendations, corroborating our assertion that recommendations for groups can be useful in various domains.

One of the "lessons learned" cited by the PolyLens article is that "better social value functions for group predictions are needed". But we have already seen that in this domain the notion of "better" is very subjective, and it is subject to the nature of the group and the problem being considered. So how can we treat this?

One possible approach is to admit that we cannot know what is better for a group, and not try to enter in this domain. This leaves us with at least two alternatives:

- The first is to furnish the users with flexible, parameterized methods of generating the recommendations and let them adjust them to meet their group's needs. This implies that the methods should be easy to interpret, with "human meaning", such that the users could understand what they are parameterizing. In Chapter 4 we propose one method that tries to fulfill these properties.
- Other alternative is to use more than one method and automatically learn which one is more appropriate for each group. This is an interesting alternative, because it would be possible to recommend for each group using the method most suitable for it. In order to use learning, it is necessary that the same group uses the system many times (which is a reasonable assumption in some do-

4 http://movielens.umn.edu

mains, like watching TV at home with the family). The "automatic adaptation" of the recommender system to the group alleviates the need of user-tunable methods, and even complex black box methods could be used.

However, as we said in Section 3.2.5, the nonexistence of a metric capable of universally quantifying the satisfaction of a group does not mean we cannot compare different alternatives. By defining metrics which reflect desirable properties, we can say that a higher score for a strategy in such metrics means that it has better performance in view of these metrics. This may be a strong indication of "superior performance" if the group uses similar criteria to the ones reflected by the metrics. In Chapter 6 we propose methods to experimentally evaluate group recommendations and in Chapter 7 we evaluate the recommendation strategies proposed in Chapters 4 and 5 using the evaluation methods proposed.

Chapter 4

Recommending for groups using aggregation-based methodologies

One way for making recommendations for groups is to build individual recommendations and aggregate them. We see in this chapter one alternative for doing this using collaborative filtering and fuzzy majority.

4.1 Overview

As cited in Chapter 2, one way to use the collaborative filtering process to recommend for groups is to first recommend to individuals, and then aggregate the results, thus reaching a recommendation for the group. This gives rise to the problem of how to aggregate the individual results in order to reach an outcome for the group.

One way to approach this problem is to have an aggregation strategy that can be adjusted by the users, so that they can tune it for their needs. Consequently, one requisite of this approach is that the strategy presents itself to the user in a comprehensible way, so that s/he can understand what is being adjusted. This fits well into the framework of fuzzy majority. By using fuzzy linguistic quantifiers that express the human discourse, fuzzy majority provides a framework with greater "human consistency" to the decision process.

In this chapter, we will first present a classification method of alternatives based on fuzzy majority proposed by Chiclana et al. [12] in the context of aggregating the opinions of multiple experts over a subject. Then we will see how this method can be used to reach a recommendation for a group from the individual recommendations given by a collaborative filtering system. An initial assessment of this method to recommend for groups was published in [46].

4.2 Fuzzy majority

Traditionally, the majority is defined by a threshold given the number of individuals. For example, for ten individuals we can set "six" as the limit. Fuzzy majority, on the other hand, is a more flexible concept, manipulated using fuzzy logic based on linguistic quantifiers.

In this section we present the fuzzy quantifiers, used to represent the concept of fuzzy majority, and the *ordered weighted averaging* (OWA) operator, used to aggregate information. The OWA operator reflect the fuzzy majority by calculating the weights used in the aggregation by means of a fuzzy quantifier.

4.2.1 Fuzzy linguistic quantifiers

Quantifiers can be used to represent the quantity of items that satisfy a given predicate. Classic logic is restricted to the use of two quantifiers: *exists* and *for all*, which are respectively related to the connectives *or* and *and*. Human discourse is much more richer and diverse in its quantifiers, for example, *around five*, *almost all*, *some*, *many*, *most*, *as many as possible*, *almost half*, *at least half* are examples of familiar quantifiers to the human discourse. To try to fill the void between human discourse and formal systems, providing a more flexible way to represent knowledge, the concept of linguistic quantifiers was introduced.

The semantic of a linguistic quantifier can by captured by using fuzzy subsets to represent it. There are two types of linguistic quantifiers: *absolute,* and *proportional* or *relative*. The absolute quantifiers are used to represent quantities that are absolute by nature, like *approximately 3* or *more than 10*. These absolute fuzzy linguistic quantifiers

are strongly related to the concept of quantity of elements. They are defined as fuzzy subsets of the nonnegative real numbers, $\Re^+$. In this way, an absolute quantifier can be represented by a fuzzy subset Q, such that for every $r \in \Re^+$ the membership degree of r in Q, Q(r), indicates the degree in which the quantity r is compatible with the quantifier represented by Q. Proportional quantifiers such as *most*, *at least half*, can be represented by fuzzy subsets in the unit interval, [0, 1]. For every $r \in [0, 1]$, Q(r) indicates the degree in which the proportion r is compatible with the semantic of this quantifier. Any natural language quantifier can be represented by proportional fuzzy quantifiers, or given the cardinality of the considered elements, by an absolute quantifier. Functionally, linguistic quantifiers in general are often of one of the types: *increasing*, *decreasing*, or *unimodal*. An increasing quantifier is characterized by the relation

$$Q(r_1) \geq Q(r_2) \text{ if } r_1 > r_2.$$

Increasing quantifiers can be used to represent terms like *at least x*, *all*, *most* etc. A decreasing quantifier is characterized by the relation

$$Q(r_1) \leq Q(r_2) \text{ if } r_1 > r_2.$$

These quantifiers can be used to express terms like *a few*, *at most x*. Unimodal quantifiers have the property

$$Q(a) \leq Q(b) \leq Q(c) = 1 \geq Q(d)$$

for some $a \leq b \leq c \leq d$. They are useful to represent terms like *about x*.

An absolute quantifier $Q : \Re^+ \rightarrow [0, 1]$ satisfies the property:

$$Q(0) = 0, \text{ and } \exists k \text{ such that } Q(k) = 1.$$

A relative quantifier $Q : [0, 1] \rightarrow [0, 1]$ satisfies the property:

$$Q(0) = 0, \text{ and } \exists r \in [0, 1] \text{ such that } Q(r) = 1.$$

The membership function of an increasing relative quantifier can be represented as:

$$Q(r) = \begin{cases} 0 & \text{if } r < a \\ \frac{r-a}{b-a} & \text{if } a \leqslant r \leqslant b \\ 1 & \text{if } r > b \end{cases} \tag{4.1}$$

with $a, b, r \in [0, 1]$.

In this work we will use the increasing relative quantifiers *most*, *at least half* and *as many as possible*, with (a, b) values (described in the literature) of (0.3, 0.8), (0, 0.5) and (0.5, 1), respectively (Figure 4.1).

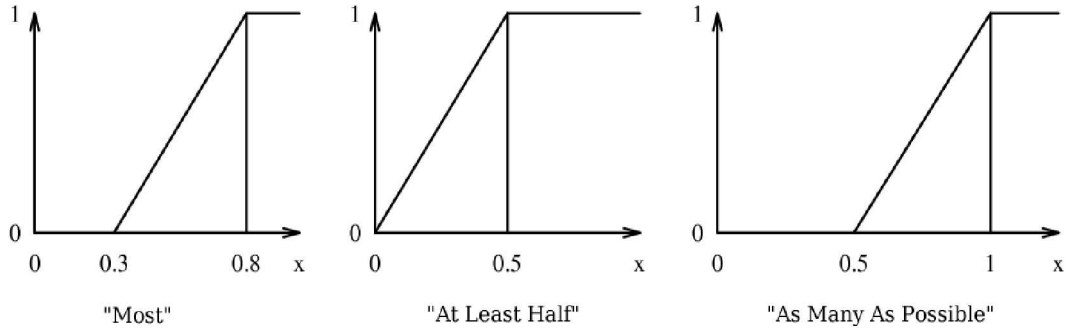


Figure 4.1 Fuzzy linguistic quantifiers

4.2.2 The OWA operator

The OWA operator provides a family of aggregation operators with the *and* operator at one extreme and the operator *or* at the other extreme.

An *n*-dimensional OWA operator is a function ϕ,

$$\phi : [0, 1]^n \rightarrow [0, 1],$$

that is associated with a vector of weights. Let {a1, ..., an} be a list of values to aggregate, then the OWA operator ϕ is defined as

$$\phi(a_1, \ldots, a_n) = W \cdot B^T = \sum_{i=1}^{n} w_i b_i \tag{4.2}$$

where $W = [w_1, ..., w_n]$ is a vector of weights, such that $w_i \in [0, 1]$ and $\sum_i w_i = 1$; B is the vector of ordered values. That is, each element $b_i \in B$ is the i^{th} larger value in the collection $a_1, ..., a_n$ (decreasing order).

The OWA operator has the maximum (*or*) at one extreme, the minimum (*and*) at the other and other intermediate values (like the mean) can be obtained by choosing appropriate weights:

- For $W = [1, 0, ..., 0]$, $\phi(a_1, ..., a_n) = \max_i a_i$ (*Or*)
- For $W = [0, 0, ..., 1]$, $\phi(a_1, ..., a_n) = \min_i a_i$ (*And*)
- For $W = [1/n, 1/n, ..., 1/n]$, $\phi(a_1, ..., a_n) = \text{avg}(a_1, ..., a_n)$ (*Mean*)

A natural question that arises is how to obtain the weights for the OWA operator. One alternative is to try to learn the weights from examples using some machine learning technique; another is to give some semantics or meaning to the weights. This latter alternative has found multiple applications on areas of fuzzy and multivalued logics, evidence theory, design of fuzzy controllers, and quantifier guided aggregations.

The interest of this method is in the area of quantifier guided aggregations. The idea is to calculate the weights for the aggregation scheme (made by means of the OWA operator) using linguistic quantifiers that represent the concept of fuzzy majority. In this way, we can give the semantics of the linguistic quantifier used to the aggregation. Yager [65] suggested an way to compute the weights of the OWA aggregation operator using fuzzy linguistic quantifiers. In the case of increasing relative quantifiers, it is given by the expression:

$$w_i = \mathrm{Q}(i/n) - \mathrm{Q}((i-1)/n),\ i = 1, \ldots, n \quad . \tag{4.3}$$

Yager showed that for any increasing relative quantifier, this formula will always get:

- $\sum_i w_i = 1$;
- $w_i \in [0, 1]$.

And when used with some basic quantifiers, like *for all*, *there exists* and *the identity quantifier* it generates the expected vectors of weights (respectively the weights for *and*, *or* and *mean*).

When a linguistic quantifier Q is used to compute the weights of an OWA operator ϕ, it will be symbolized by ϕ_Q.

4.3 The decision process: classification method of alternatives

It is assumed that there is a finite set of alternatives $X = \{x_1, ..., x_n\}$ as well as a finite set of experts $E = \{e_1, ..., e_m\}$. Each expert $e_k \in E$ provides his/her opinion about X as an individual preference ordering $\{x_{o(1)}, ..., x_{o(n)}\}$, where $o(\cdot)$ is the permutation function over the set of subscripts $\{1, ..., n\}$. Each expert classify the alternatives according to a weak ordering[5] from the best to the worst alternative.

The decision process embodies two steps. In the first step, named *aggregation*, the individual preference orderings are aggregated in order to devise a collective fuzzy preference relation. In the second step, named *exploitation*, the collective fuzzy preference relation is used to obtain a global ranking of the alternatives. In the next two sections, these steps are presented.

4.3.1 Aggregation: obtaining the collective preference ordering relation

For each individual preference ordering a preference relation P^k is derived, where p^k_{ij} reflects the preference over the alternatives x_i and x_j for the expert e_k, $p^k_{ij} \in \{0, 1\}$. It assumes the value 1 if x_i is preferred to x_j, and 0 otherwise. In this way we have a collection of binary preference relations:

$$\{P^1, ..., P^m\}.$$

5 A weak ordering is complete, transitive and reflexive. It is not assymetric (that is, "ties" are allowed)

From the set of binary preference relations the collective preference relation P will be obtained. This will be done by means of an OWA operator, with its weights based on a linguistic quantifier.

Two possibilities of aggregation in respect to the intensity of the experts' opinions are considered. In the first case they are assumed to be all equal (homogeneous case), whereas in the second they can have different importances (heterogeneous case). The latter is a generalization of the first.

Each value $p_{ij} \in [0, 1]$ of P will represent the degree to which the affirmative "alternative x_i is at least as good as alternative x_j" is true.

Aggregation with homogeneous experts

In this case the opinions of the experts are taken to have the same intensity. The individual preference relations $\{p^1_{ij}, ..., p^m_{ij}\}$ are aggregated to obtain p_{ij}, for every i, j. This is done using fuzzy majority. By means of the fuzzy quantifier chosen for this phase, the vector of weights of the OWA operator is calculated (using Equation 4.3). The OWA operator is then used to obtain the collective preference relation P as

$$P = \phi_Q(P_1, ..., P_m)$$

where $p_{ij} = \phi_Q(p^1_{ij}, ..., p^m_{ij})$, and the aggregation is done using Equation 4.2.

Aggregation with heterogeneous experts

In this case, associated with the experts we have their respective importance degrees as a fuzzy subset, such that, $\mu_E(k) \in [0, 1]$ denotes the importance degree for the expert e_k.

Assuming that in this context each value $\mu_E(k)$ is a weight that indicates the importance of the expert in the aggregation process, the general procedure to include the importance in the aggregation involves the transformation of the preference values under the importance degrees. This transformation follows the expression:

$$p^{k'}_{ij} = g(p^k_{ij}, \mu_E(k)).$$

As a default operation for g we can use the Min aggregation operator, which is the default fuzzy implementation for the intersection of fuzzy sets. Therefore we have,

$$p^{k'}_{ij} = Min\{p^k_{ij}, \mu_E(k)\}.$$

When all experts have the same importance ($\mu_E(k) = 1$ for every $k \in \{1, ..., n\}$), $p^{k'}_{ij}$ is reduced to p^k_{ij}.

4.3.2 Exploitation: ranking the alternatives from the collective preference relation

At this point, in order to select the "best" alternatives to the group of individuals, two quantifier guided choice degrees of alternatives based on the concept of fuzzy majority are used: a dominance degree and a non-dominance degree. Both are based on the use of the OWA operator with its weights calculated by means of the quantifier used to represent the fuzzy majority at this phase.

Quantifier guided dominance degree

The quantifier guided dominance degree (QGDD) is used to quantify the dominance degree that one alternative has over all others from the point of view of the fuzzy majority. The QGDD of an alternative x_i is calculated as:

$$\mathrm{QGDD}(x_i) = \phi_Q(p_{ij} \mid j \in \{1, \ldots, n\} \text{ and } j \neq i) \tag{4.4}$$

where ϕ_Q is an OWA operator with weights defined by the linguistic quantifier Q, and whose components (to aggregate) are the elements of the corresponding row of P, that is, for x_i, the set of n - 1 values $\{p_{ij} \mid j \in \{1, ..., n\} \text{ and } j \neq i\}$.

The elements of the set

$$X^{\mathrm{QGDD}} = \{x \mid x \in X, \mathrm{QGDD}(x) \geq \mathrm{QGDD}(z), \text{for every } z \in X\}$$

are called maximal dominance elements of the fuzzy majority of X quantified by Q.

Quantifier guided non-dominance degree

The quantifier guided non-dominance degree (QGNDD) expresses the degree to which one alternative is not dominated by a fuzzy majority of the others. It is defined by the expression:

$$\mathrm{QGNDD}(x_i) = \phi_Q(1 - p^s_{ji} \mid j \in \{1, \ldots, n\} \text{ and } j \neq i) \tag{4.5}$$

where

$$p^s_{ji} = \max\{p_{ji} - p_{ij}, 0\}$$

expresses the degree to which x_i is strictly dominated by x_j.

The elements of the set

$$X^{\mathrm{QGNDD}} = \{x \mid x \in X, \mathrm{QGNDD}(x) \geq \mathrm{QGNDD}(z), \text{for every } z \in X\}$$

are called maximal non-dominated elements by the fuzzy majority of X quantified by Q.

Selection process

The degrees of dominance and non-dominance can be used to choose the best alternatives in one of the following ways:

- **Sequential selection process**: choose one of the two choice degrees and use it to obtain the set of maximal elements. If there is more than one element in this set, the other degree can be used as a second selection criterion.
- **Conjunction selection process**: apply the two choice degrees, obtaining the sets X^{QGDD} and X^{QGNDD}. The final selection is the intersection of these two sets. Notice that this selection process is more restrictive and can result in an empty set.

Figure 4.2 schematically represents the decision process using fuzzy majority.

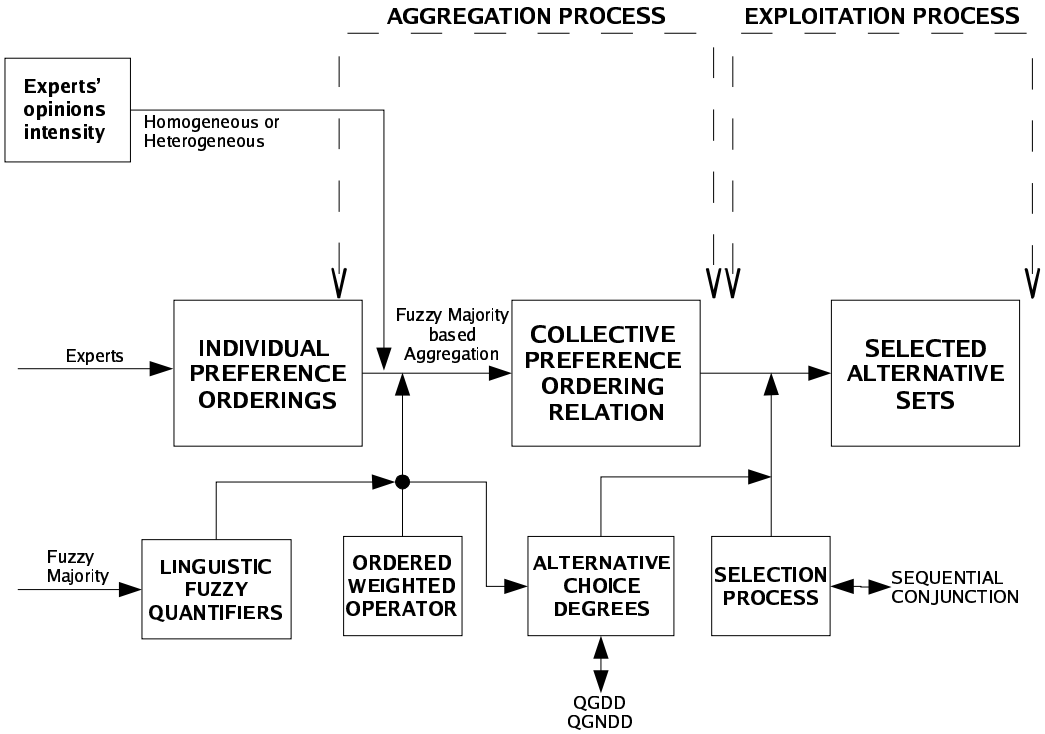


Figure 4.2 Process of classificating alternatives based on fuzzy majority

4.4 Example: obtaining recommendations for a group using the fuzzy method

Let's see an example where we have a group of 6 people ($m = 6$) and the set of items that can be recommended is $X = \{x_1, x_2, x_3, x_4\}$. We will consider that every person has the same importance (homogeneous case).

The first step is to use collaborative filtering for each person, so that we have their predicted grades for each of the items. We can do this using the methodology described in Section 2.3.1.

Let's suppose that the collaborative filtering process has supplied the following grades:

$$G^1 = (4.7, 3.8, 3.5, 2.1)$$

$$G^2 = (4.8, 4.0, 1.1, 3.2)$$
$$G^3 = (2.2, 2.5, 1.8, 1.4)$$
$$G^4 = (3.2, 4.8, 2.3, 3.0)$$
$$G^5 = (2.4, 2.0, 4.6, 1.5)$$
$$G^6 = (3.0, 2.2, 4.1, 2.4)$$

where G^k signifies the grades predicted for the individual k. The first grade refers to item x_1, the second to item x_2 and so on.

Now we begin the aggregation phase. First, we obtain the set of individual preference relations, $\{P^1, ..., P^m\}$:

P^1	x_1	x_2	x_3	x_4
x_1	-	1	1	1
x_2	0	-	1	1
x_3	0	0	-	1
x_4	0	0	0	-

P^2	x_1	x_2	x_3	x_4
x_1	-	1	1	1
x_2	0	-	1	1
x_3	0	0	-	0
x_4	0	0	1	-

P^3	x_1	x_2	x_3	x_4
x_1	-	0	1	1
x_2	1	-	1	1
x_3	0	0	-	1
x_4	0	0	0	-

P^4	x_1	x_2	x_3	x_4
x_1	-	0	1	1
x_2	1	-	1	1
x_3	0	0	-	0
x_4	0	0	1	-

P^5	x_1	x_2	x_3	x_4
x_1	-	1	0	1
x_2	0	-	0	1
x_3	1	1	-	1
x_4	0	0	0	-

P^6	x_1	x_2	x_3	x_4
x_1	-	1	0	1
x_2	0	-	0	0
x_3	1	1	-	1
x_4	0	1	0	-

In order to obtain the collective preference relation, P, we will choose to use (to illustrate) the linguistic quantifier *as many as possible*, that has a = 0.5 and b = 1 (see Figure 4.1). First we will calculate the weights for the OWA operator based on the quantifier we just chosen (Equations 4.3 and 4.1):

$$w_1 = Q(1/6) - Q(0) = 0 - 0 = 0$$
$$w_2 = Q(2/6) - Q(1/6) = 0 - 0 = 0$$
$$w_3 = Q(3/6) - Q(2/6) = 0 - 0 = 0$$
$$w_4 = Q(4/6) - Q(3/6) = 0.33 - 0 = 0.33$$
$$w_5 = Q(5/6) - Q(4/6) = 0.67 - 0.33 = 0.34^6$$
$$w_6 = Q(6/6) - Q(5/6) = 1 - 0.67 = 0.33$$

6 Notice that if we had used full precision in these calculations, the weights w_4, w_5 and w_6 would be $0.33\overline{3}$.

The OWA operator is now used to calculate each p_{ij} of P:

$$p_{12} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 1, 1, 0, 0]^T = 0.33$$
$$p_{13} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 1, 1, 0, 0]^T = 0.33$$
$$p_{14} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 1, 1, 1, 1]^T = 1.00$$
$$p_{21} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 0, 0, 0, 0]^T = 0.00$$
$$p_{23} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 1, 1, 0, 0]^T = 0.33$$
$$p_{24} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 1, 1, 1, 0]^T = 0.67$$
$$p_{31} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 0, 0, 0, 0]^T = 0.00$$
$$p_{32} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 0, 0, 0, 0]^T = 0.00$$
$$p_{34} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 1, 1, 0, 0]^T = 0.33$$
$$p_{41} = [0, 0, 0, 0.33, 0.34, 0.33] \times [0, 0, 0, 0, 0, 0]^T = 0.00$$
$$p_{42} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 0, 0, 0, 0, 0]^T = 0.00$$
$$p_{43} = [0, 0, 0, 0.33, 0.34, 0.33] \times [1, 1, 0, 0, 0, 0]^T = 0.00$$

Representing P in matrix form, we have:

P	x_1	x_2	x_3	x_4
x_1	-	0.33	0.33	1.00
x_2	0.00	-	0.33	0.67
x_3	0.00	0.00	-	0.33
x_4	0.00	0.00	0.00	-

Now we proceed with exploitation phase. For each item we will calculate the QGDD and the QGNDD. To illustrate, we will use the linguistic quantifier *most* to calculate both degrees, which have a =0.3 and b = 0.8 (see Figure 4.1). In this phase the OWA operator will be used to aggregate 3 values (each item in relation to the others) whereas in the first phase we used it to aggregate 6 values (number of persons). Calculating the weights for the OWA operator based on the chosen quantifier (Equations 4.3 and 4.1):

$$w_1 = Q(1/3) - Q(0) = 0.07 - 0 = 0.07$$
$$w_2 = Q(2/3) - Q(1/3) = 0.73 - 0.07 = 0.66$$
$$w_3 = Q(3/3) - Q(2/3) = 1 - 0.73 = 0.27$$

Calculating the QGDD for the items (Equation 4.4):

$$\begin{aligned}
\text{QGDD}(x_1) &= [0.07, 0.66, 0.27] \times [1.00, 0.33, 0.33]^T = 0.38 \\
\text{QGDD}(x_2) &= [0.07, 0.66, 0.27] \times [0.67, 0.33, 0.00]^T = 0.26 \\
\text{QGDD}(x_3) &= [0.07, 0.66, 0.27] \times [0.33, 0.00, 0.00]^T = 0.02 \\
\text{QGDD}(x_4) &= [0.07, 0.66, 0.27] \times [0.00, 0.00, 0.00]^T = 0.00
\end{aligned}$$

Calculating the QGNDD (Equation 4.5):

- x_1:

  $p^s_{21} = \max(0.00 - 0.33, 0) = 0.00$, $p^s_{31} = \max(0.00 - 0.33, 0) = 0.00$, $p^s_{41} = \max(0.00 - 1.00, 0) = 0.00$.

  Values to aggregate: {1.00, 1.00, 1.00}.

  $\text{QGNDD}(x_1) = [0.07, 0.66, 0.27] \times [1.00, 1.00, 1.00]^T = 1.00$.

- x_2:

  $p^s_{12} = \max(0.33 - 0.00, 0) = 0.33$, $p^s_{32} = \max(0.00 - 0.33, 0) = 0.00$, $p^s_{42} = \max(0.00 - 0.67, 0) = 0.00$.

  Values to aggregate: {0.67, 1.00, 1.00}

  $\text{QGNDD}(x_2) = [0.07, 0.66, 0.27] \times [1.00, 1.00, 0.67]^T = 0.92$.

- x_3:

  $p^s_{13} = \max(0.33 - 0.00, 0) = 0.33$, $p^s_{23} = \max(0.33 - 0.00, 0) = 0.33$, $p^s_{43} = \max(0.00 - 0.33, 0) = 0.00$.

  Values to aggregate: {0.67, 0.67, 1.00}.

  $\text{QGNDD}(x_3) = [0.07, 0.66, 0.27] \times [1.00, 0.67, 0.67]^T = 0.69$.

- x_4:

  $p^s_{14} = \max(1.00 - 0.00, 0) = 1.00$, $p^s_{24} = \max(0.67 - 0.00, 0) = 0.67$, $p^s_{34} = \max(0.33 - 0.00, 0) = 0.33$.

  Values to aggregate: {0.00, 0.33, 0.67}.

  $\text{QGNDD}(x_4) = [0.07, 0.66, 0.27] \times [0.67, 0.33, 0.00]^T = 0.26$.

Summarizing the obtained results:

	x_1	x_2	x_3	x_4
QGDD	0.38	0.26	0.02	0.00
QGNDD	1.00	0.92	0.69	0.26

This values represent—for the QGDD—the dominance degree that each alternative has over *most* (quantifier used in the second phase) of the others according to *as many as possible* (quantifier used in the first phase to aggregate opinions) persons in the group; and, for the QGNDD, the degree to which each alternative is not dominated by *most* of the others according to *as many as possible* persons.

The maximal sets clearly are:

$$X^{\text{QGDD}} = \{x_1\} \text{ and } X^{\text{QGNDD}} = \{x_1\},$$

consequently for both selection processes x_1 would be considered the best alternative.

However when we are making a recommendation, we do not need to be so rigid and only suggest the maximal alternatives. Most times the interest is to rank the alternatives, or choose the k best alternatives, where k is the number of suggestions that will be given. In this context, we can elect one of the two choice degrees (QGDD or QGNDD) to order the alternatives, and then recommend the best k. The other choice degree can be used to break ties in this situation.

For example, if we wanted to rank the alternatives $\{x_1, x_2, x_3, x_4\}$ we could use the QGDD for this, and the QGNDD to break ties. In the example presented, there are no ties and the rank obtained using the QGDD or the QGNDD is the same: (x_1, x_2, x_3, x_4).

In the experiments conducted in Chapter 7 we use the QGDD as the criterion to order the alternatives, and the QGNDD is used to break ties.

4.4.1 Using heterogeneous aggregations to enrich the recommendation process

We have just seen an example that uses fuzzy aggregation to generate recommendations for groups. In this example we used homogeneous aggregations (see Section 4.3.1), where every group member is given the same importance. Nevertheless, heterogeneous aggregations can be used to introduce different characteristics to the recommendation process.

Not only heterogeneous aggregations can be used to offer the user the opportunity to give different levels of importance for each group member, but they could also be used by the recommender system to bias the results according to some interesting criterion. For example, a recommender system could use some of the following criteria to weight differently the members of a group:

- **Number of evaluations**: the recommender system could consider that users with more evaluations are "experts" and more importance should be given to their opinions. Consequently these "experts" will have higher weights than the other users in the group who have less evaluations.
- **Collaborative filtering quality**: a recommender system can have an idea of how good a given recommendation (for an individual) is. One way to do this is to take account of the neighbors used to generate the recommendation. If they share a large set of items evaluated in common with the user (for whom the system is recommending) and have a high correlation to him/her, the recommend-

ation will be on a more solid basis. Otherwise, it will have a greater chance of being weak. Therefore, a recommender system could use as one criterion to weight the individuals in the aggregation the probable quality of the individual recommendations for each user.

- **Decision strategy**: as seen in Section 3.2.3, a variety of different decision schemes have been used to model the behavior of groups. We could give the opportunity to the user receive the recommendation under different decision schemes using the fuzzy majority methodology. For this, the members of the group would be weighted accordingly. For example, inspired by decisions schemes influenced by distance we could weight each user by how far their predicted grades are on average from the mean predicted grade.
- **Historical fairness**: considering that a recommender system can repeatedly give recommendations for the same group (for example, a family), it could weight the users in such a way that members of the group that were more dissatisfied with previous recommendations receive a higher weight this time (and vice-versa). In this way it would try to avoid that the same group members were frequently dissatisfied with the recommendations, a situation that could encourage them to leave the group (or not use the recommender system anymore).

Chapter 5

Recommending for groups using model-based methodologies

An approach for making recommendations for groups is to represent the group by means of a model and recommend to this model. In this chapter we will develop a model-based recommendation methodology for groups.

5.1 Overview

In the last chapter, an aggregation-based strategy to recommend for groups has been shown. This kind of approach first recommends for the individuals in the group and then aggregates these recommendations to yield the final group recommendation. In this chapter, we work on another approach to the problem: model-based strategies. Now, instead of recommending for individuals and then aggregating the results, we will first build a model to represent the group and recommend directly to this model.

We begin by introducing the topic of *symbolic data analysis*. This topic will be the basis for a novel model-based strategy to recommend for groups we present subsequently. We finish by comparing the features of aggregation-based strategies and model-based ones.

5.2 Symbolic data analysis

Nowadays, at every moment, a large quantity of data is being recorded. A task of fundamental importance is to extract the underlying concepts embodied in these data. To describe these concepts, more powerful data tables are needed, that can accommodate cells with more complex data types. Each cell may contain not only a single quantitative or categorical value, but data of different types [15]:

- Single quantitative value;
- Single categorical value;
- Multivalued data. For example color(w) = {*red*, *green*, *blue*} meaning that the color of w may be *red*, *green* or *blue*;
- Interval data. For example weight(w) = [50, 150];
- Multivalued data with weights (a histogram or membership function);

The variables may be taxonomic ("the color is considered hot if it is yellow, orange or red"), hierarchical dependent (the variables "do you have a car?" and "brand of the car" are hierarchically linked), or with logical dependences ("if age(w) is less than 2 then weight(w) is less than 30").

This richer type of data is called *symbolic data*. The development of new methods of data analysis suitable to treat this type of data (or the extension of existing methods to this type of data) is the aim of *symbolic*[7] *data analysis* [7].

In the next section we develop a model-based approach for group recommendations that represents groups of persons and items by means of symbolic data.

5.3 A symbolic approach for making group recommendations

In recommender systems that use neighborhood-based collaborative filtering (like the example showed in Section 2.3.1), the complexity to make recommendations grows with the number of users. Many of these systems must give on-line responses,

7 Notice that the term "symbolic" in symbolic data analysis has no relation with its meaning in "symbolic artificial intelligence", it is just a nomenclature clash.

and may have thousands of users. This makes almost impossible to search for neighbors on-line, using the whole set of users. To tackle this problem, some recommender systems (for individuals) search for neighbors only among users that have recently used the system and are still in a small primary memory cache. This, of course, may result in degradation and fluctuation of the system performance. Other more elaborate approaches have been proposed (in the domain of recommender systems for individuals). Sarwar et al. [54], for example, developed a method based on the "neighborhood" of items, instead of users. Because the relationships between items are relatively static, data may be pre-computed, requiring less on-line computation.

In this section we develop a model-based recommendation strategy for groups. It builds a model for the group and recommends directly for it, dispensing with the need of generating recommendations for each group member and aggregating them afterwards. During the recommendation process, it uses models for the items—which can be pre-computed—and does not require the computation of on-line user neighborhoods.

The intuition behind our approach is that for each item we can identify the group of people who like it and the group of people that do not like it. We assume that the group for which we will make a recommendation will appreciate an item if the group has similar preferences to the group of people who like the item and is dissimilar to the group of people who do not like it.

To implement this, first the group of users for whom the recommendations will be computed is represented by a prototype that contains the histogram of rates for each item evaluated by the group. The target items (items that can be recommended) are also represented in a similar way, but now we create two prototypes for each target item: a positive prototype, that contains the histogram of rates for (other) items evaluated by individuals who liked the target item; and a negative prototype that is analogous to the positive one, but the histogram of rates is from individuals who did not like the target item. Next we compute the similarity between the group prototype and the two prototypes of each target item. The final similarity between a target item i and a group g is given by

$$\mathrm{sim}(g, i) = \frac{\mathrm{protsim}(\mathrm{gpt}(g), \mathrm{pospt}(i)) + 1 - \mathrm{protsim}(\mathrm{gpt}(g), \mathrm{negpt}(i))}{2} \tag{5.1}$$

where gpt(g) is the prototype for the group g, pospt(i) is the positive prototype for target item i, negpt(i) is the negative prototype for target item i, and protsim($\cdot$, $\cdot$) is a function that takes a group prototype and an item prototype as arguments and returns a similarity value v between them, $v \in [0, 1]$.

Finally, we order the target items by decreasing order of similarity values. If we want to recommend k items to the users, we can take the first k items of this ordering. Figure 5.1 depicts the recommendation process.

The two aspects of this methodology, the creation of prototypes and the similarity computation will be described in the following subsections.

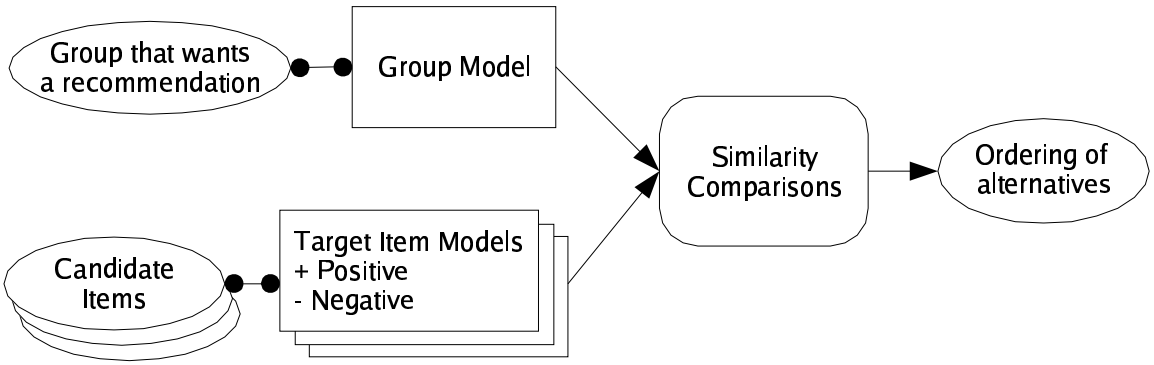


Figure 5.1 Recommendation process

5.3.1 Prototype generation

A fundamental step of this method is the prototype generation. The basic idea is that the group and the target items are represented by the histograms of rates for items. Furthermore, different weights can be attributed to each histogram that make up the prototypes. In other words, each prototype is described by a set of p symbolic variables Y_j. Each item corresponds to a categorical modal variable Y_j that may also have an associated weight. The modalities of Y_j are the different grades that can be given to items. In our case, we have six modalities. Figure 5.2 shows a prototype described by 3 items (categorical modal variables).

Group prototype

In the group prototype we have the grade histograms for every item that has been evaluated by at least one member of the group. The grade histogram is built by computing the frequency of each modality in the ratings of the group members for the item being considered. The used data has a discrete set of 6 grades (thus 6 modalities): {0.0, 0.2, 0.4, 0.6, 0.8, 1.0}, where 0.0 is the worst grade and 1.0 is the best. For example, if an item i_1 was evaluated by 2 users in a group of 3 individuals and they gave the ratings 0.4 and 0.6 for the item, the row in the symbolic data table corres-

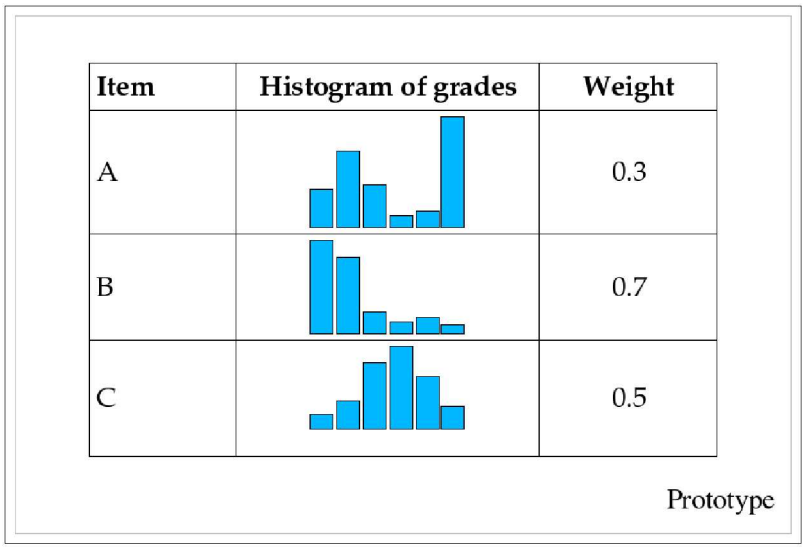


Figure 5.2 Example of a prototype (of a group or one of the prototypes of a target item)

ponding to the item (see the prototype structure in Figure 5.2) would be: $\{i_1, \{0.0, 0.0, 0.5, 0.5, 0.0, 0.0\}, 0.667\}$, assuming the weight as the fraction of the group that has evaluated the item.

Prototypes of the target items

To build a prototype for a target item, the first step is to decide which users will be selected to have their evaluations in the prototype. This users have the role of characterizing the profile of those who like the target item, for the positive profile; and of characterizing the profile of those who do not like the target item, for the negative profile.

Therefrom, for the positive prototype only the users that evaluated the target item highly are chosen. Users that have given grades 0.8 or 1.0 were chosen as the "positive representatives" for the group. For the negative prototype the users that have given 0.0 or 0.2 for the target item were chosen.

One parameter for the building of the models is how many users will be chosen for each target item. We tested 30, 50, 100, 200 and 300 users. The preference ordering to choose these users was the grade given to the target item (that is, users that have given 1.0 to the item are preferred over those who have given 0.8 in the building of the positive prototype; and analogously, for the negative group, those that have given 0.0 are preferred over those who have given 0.2). The number of evaluations was used as a second criterion for ordering the users, giving preference to users with more evaluations. The rationale behind giving preference to users with more evaluations is to build a "richer" prototype for the target item.

5.3.2 Similarity calculation

To compute the similarity between the prototype of a group and the prototype of a target item, we only consider the items that are in both prototypes (Figure 5.3). As similarity measure to compute the similarity between the prototypes we tried Bacelar-Nicolau's weighted affinity coefficient (presented in [7]) and two measures based on the Euclidean distance and the Pearson correlation, respectively.

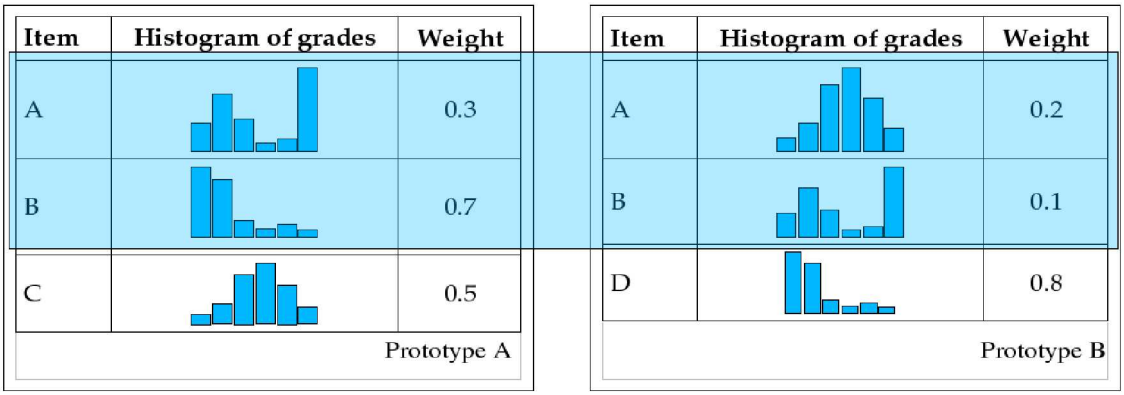


Figure 5.3 When comparing two prototypes, only items available in both of them are considered. In the example, only the data about items A and B will be compared.

Similarity measures

Euclidean Distance

The similarity between two prototypes k and k′ based on the Euclidean distance is given by:

$$\begin{aligned} \text{protsim}(k, k') &= \sum_{j=1}^{p} w_j \cdot (1 - c_f \,\text{dist}(E_{kj}, E_{k'j})) \\ &= \sum_{j=1}^{p} w_j \cdot \left(1 - c_f \sqrt{\sum_{l=1}^{m_j} \left(n_{kjl} - n_{k'jl}\right)^2}\right) \end{aligned} \tag{5.2}$$

where:

- p is the number of items present in both prototypes;
- w_j is the weight attributed to item j;
- m_j is the number of modalities (six, corresponding to the six different rates);
- n_{kjl} and $n_{k'jl}$ are the relative frequencies obtained by rate l in the prototypes k and k' for the item j, respectively.
- c_f is a normalization factor to guarantee that the computed distance stays in the interval [0, 1]

Correlation

The similarity between two prototypes k and k' using Pearson correlation is given by:

$$\text{protsim}(k, k') = \sum_{j=1}^{p} w_j \cdot \frac{\left(1 + \text{correlation}(E_{kj}, E_{k'j})\right)}{2} \quad \text{, where}$$

$$\text{correlation}(E_{kj}, E_{k'j}) = \frac{\sum_{l=1}^{m_j} (n_{kjl} \cdot n_{k'jl}) - \frac{\sum_{l=1}^{m_j} n_{kjl} \sum_{l=1}^{m_j} n_{k'jl}}{m_j}}{\sqrt{\left(\sum_{l=1}^{m_j} (n_{kjl}^2) - \frac{\left(\sum_{l=1}^{m_j} n_{kjl}\right)^2}{m_j}\right)\left(\sum_{l=1}^{m_j} (n_{k'jl}^2) - \frac{\left(\sum_{l=1}^{m_j} n_{k'jl}\right)^2}{m_j}\right)}} \tag{5.3}$$

with the variables defined as before.

Bacelar-Nicolau's Affinity Coefficient

The affinity coefficient is given by:

$$\text{protsim}(k, k') = \sum_{j=1}^{p} w_j \cdot aff(E_{kj}, E_{k'j}) = \sum_{j=1}^{p} w_j \cdot \sum_{l=1}^{m_j} \sqrt{n_{kjl} \cdot n_{k'jl}} \tag{5.4}$$

with the variables defined as before.

This coefficient gives a number between 0 and 1, with value 1 if k and k' are identical or proportional and 0 if they are orthogonal.

Weights w_j

We have experimented several different options for the weights w_j. All weights were normalized, such that $\sum_{j=1}^{p} w_j = 1$. Table 5.1 shows the options explored and their rationales.

Table 5.1 Weights tested and their rationales.

Weight adopted	*Rationale*
No weights.	The histograms alone are adequate to compute the similarities. No weights are needed.
Maximum between the fraction of the group that has evaluated the item in the group prototype and the fraction in the prototype of the target item.	If one item was frequently evaluated by the group that wants the recommendation or by the group that represents the target item it is important ("Or" meaning of the maximum in the fuzzy domain).
Minimum between the fraction of the group that has evaluated the item in the group prototype and the fraction in the prototype of the target item.	An item is important only if it has been frequently evaluated by the group that wants the recommendation and the group that represents the target item ("And" meaning of the minimum in the fuzzy domain).
Entropy[8] of the histogram in the prototype of the group.	When there is concordance in the group that wants the recommendation about this item, it is more important.
Standard deviation of the histogram in the prototype of the group.	Idem.
Entropy of the histogram in the prototype of the target item.	When there is concordance in the group that represents the item about this item, it is more important.
Standard deviation of the histogram in the prototype of the target item.	Idem
Similarity between the target item and the current item being considered in the prototype.	When the current item is more similar to the target item, the opinion about it is more important.

Other adjustments to the model

As another adjustment to the model, we experimented to give more importance to the extremes of the distributions (supposing that the extreme grades are more important to differentiate the items). For this, we tested two deformations. In the first, the number of occurrences of the modalities was multiplied by the factors {3, 2, 1, 1, 2, 3}, respectively to the modalities (ratings) {0.0, 0.2, 0.4, 0.6, 0.8, 1.0}. In the second, the factors were {3, 3, 2, 1, 2, 3}.

8 In this case the weight was calculated as $1 + n_{kjl} \sum_{l=1}^{m_j} \log_{mj}(n_{kjl})$

Other adjustment experimented was to consider the number of items in common between the prototypes (i.e. how many comparisons were made). The rationale behind this is that similarities computed using prototypes with many items in common are more trustful than those computed using prototypes with few items in common. Therefore, the first should be promoted while the latter should be penalized. For this, we tested the following factor to multiply the final similarity between the prototypes of the target item and the group:

- $e^{x/I-1}$

where,

for I we tested the size of the mean intersection, the maximum intersection and the median intersection (considering all prototypes of the target items in relation to the group prototype); x is the size of the intersection between the current item prototype and the group prototype.

In Chapter 7 we experiment with some configurations of this recommendation model in order to try to tune its performance (using a training set), and after we will run the chosen configuration using a test set, comparing it to recommendations made by means of aggregation-based methods. The next chapter presents the design and metrics we use in these experiments.

5.4 Aggregation-based and model-based strategies compared

Different approaches to solve a problem often also bring different sets of features for each approach. This is the case when we compare the aggregation-based and model-based approaches. The following different competences can be identified when comparing the two approaches:

- **Explanation of recommendations**: recommendations generated through aggregation-based strategies allow us to explain them more easily than recommendations that used model-based strategies. In the first case, as we have the individual predictions, we can explicitly say: "This movie will satisfy Alice and Bob, but not Jim". We can even show the group the individual predicted grades (approach taken by PolyLens [44]). Using model-based strategies we do not have this possibility (saying "this movie fits the model I built for the group" will not make much sense for the users). Notice, however, that this level of explanation may not be enough, it has said nothing to explain how the individual explanations were generated in the first place. In [21], Herlocker et al. have worked on the problem of explaining individual recommendations. Initiatives like showing the users how the neighbors rated the item being recommended, how close these neighbors are from him were found valuable by users. However, showing this information for each individual in a group will most undoubtedly be a flood of information for the users, especially for larger groups. This may favor the use of *black box explanations*, which use information completely outside of the recommendation process to explain the recommendations. For example, a black box explanation may be simply an indication of past performance: "the system has been correct 75% of time when recommending

items like this". Herlocker et al. also found evidence that this kind of explanation was valuable to the users. As black box explanations are independent of the process used to come up with the recommendations, they can be used to explain recommendations generated by model-based and aggregation-based strategies.

- **Recommendation of serendipitous items**: one of the good features of collaborative filtering is its capability of finding serendipitous items to recommended. These "good surprises" are a nice feature to have in many domains. However, using aggregation-based methods we have a high chance of losing this property. By aggregating multiple collaborative filtering recommendations, we will mostly likely be favoring items that have been strongly suggested for various people. This may imply the recommendation of "obvious" items. For example, in the movie recommendation domain, the system may almost always recommend a "classic" movie to the group (and by being a classic it is probably already known by the group, configuring an useless recommendation). Model-based strategies, on the other hand, by creating a model to the group and recommending for this model have more chance of preserving serendipity. However it may also happen that a recommended item does not please a substantial portion of the group.
- **Performance for large groups**: aggregation-based strategies are potentially slower than model-based strategies for large groups, as the first have to calculate n recommendations (for a group of size n) and the latter only have to calculate one recommendation, whatever be the group size (assuming that the cost to build the group model is small).

Chapter 6

Experimental Design and Evaluation Metrics

In order to measure the quality of group recommendations, it is important to quantify their quality over groups of different characteristics. This chapter presents the experimental design and evaluation metrics used to measure the quality of the recommendations.

6.1 Overview

Groups of people have diverse characteristics: they can be big or small; made of like-minded individuals or individuals with divergent opinions. Therefore it is important to evaluate group recommendation strategies under this contrasting circumstances, i.e. different values for the variables:

- **Group size**: what is the influence of the group size on the behavior of each recommendation strategy?
- **Homogeneity degree**: what importance does the affinity among people in the group have in the performance of each recommendation strategy?

In this chapter, the experimental design and metrics that were used to experimentally evaluate the recommendation strategies described in the chapters 4 and 5 for different levels of the above mentioned variables are described.

6.2 Choice of experiments

An ideal methodology to experimentally evaluate a set of recommendation strategies for groups would be to use a two-step process. First perform off-line experiments that use historical data of a real recommender system to simulate the strategies; next perform on-line experiments to control for biases included in the data used by the off-line studies and, more importantly, to verify the effectiveness of the proposed strategies in practice. This is akin to the two-step process successfully used by Rashid et al. [47] in the context of evaluating different strategies for selecting items to be presented to new users of a recommender system (for individuals).

Obviously, to perform the on-line experiments, we need access to a running recommender system with a sufficiently large user base. The fact that the recommendations are made for groups still aggravates the need of a large community (for example, PolyLens—Section 3.3.3—even having access to the large MovieLens community was not able to achieve statistical significance in various of its observed results). Since we did not have access to such a system, we have used only off-line experiments to evaluate the strategies. One possible disadvantage of using only off-line experiments is that biases in the used data may alter the results for or against particular approaches. Section 6.5 discusses the biases we have identified.

For our study, we used a subset of the publicly available *Eachmovie* dataset, described in the next section.

6.3 The *Eachmovie* Dataset

To run our experiments, we used the *Eachmovie* dataset. *Eachmovie* was a recommender service that run for 18 months (until September, 1997) as part of a research project at the *Compaq Systems Research Center*[9]. During that period, 72,916 users gave 2,811,983 evaluations to 1,628 different movies. Users' evaluations were registered using a 6-level numerical scale (0.0, 0.2, 0.4, 0.6, 0.8, 1.0) corresponding to users eval-

9 At that time, Digital Equipment Corporation (DEC) Systems Research Center.

uations from 0 to 5 stars. The dataset is available for non-commercial use, and can be obtained from *Compaq Computer Corporation* [13]. *Eachmovie* dataset has become a "natural choice" when one needs to run simulations in recommender systems studies, especially when collaborative filtering is involved (see e.g. [6], [9], [35], [40]).

Although evaluations from 72,916 users were available, we restricted our experiments to users that had supplied at least 150 evaluations. This left 2,551 users. The cutoff of 150 is high and somewhat arbitrary. However, we needed a large number of evaluations for each user, both because the most trustful manner to see how much an user would like a given movie (when testing) is to look at the evaluation the movie received; and to have an intersection of reasonable size (of movies evaluated in common) between every pair of user. The latter characteristic grants more trustworthiness on the homogeneity degree of the groups we created. The choice of a subset of the *Eachmovie* database is commonplace in experiments that use it (see e.g. [6], [35], [47]).

For the experiments we randomly separated the movies and their corresponding evaluations into three groups: the *profile set* (50% of the movies), a *training set* (25% of the movies), and the *test set* (25% of the movies).

The next section explains how we have built different user groups based on the *Eachmovie* data. To build the groups, only evaluations about movies from the *profile set* were considered.

The movies from the test set are used to measure the quality of the recommendation. The methodology used will be described in Section 6.7.

The training set is used to adjust parameters in recommendation strategies that need it. For example, the model-based strategy of Chapter 5 has many parameters that can be varied to find an "optimal" configuration. In this case, we use the same evaluation process described in Section 6.7 but using movies from the training set instead of the test set to tune the parameters of the model. Therefore, the final evaluation (which uses the test set) is really performed using data not previously used in any step.

6.4 Data preparation: the creation of groups

In order to run the experiments, it was necessary to have groups of users with different sizes and homogeneity degrees. There is no notion of groups of users in the *Eachmovie* database, therefore, as a first step, it was necessary to create the groups.

Four levels were defined for the variable *group size*: **3, 6, 12 and 24 people**. In this way we have encompassed from very small groups (3 persons) to somewhat large groups (24 persons). We believe that this range of sizes includes most scenarios whereupon group recommendations will be useful. For the variable *homogeneity degree*, three levels were defined: **high homogeneity, medium homogeneity, and low homogeneity**. **One hundred groups were created for each combination "group size" × "homogeneity degree"**. In our context, the groups do not need to be disjoint, i.e. each individual can be in more than one group[10]. The methodology we used to build the groups is discussed in the next sections.

6.4.1 The homogeneity degree of a group

The homogeneity of a group is a subjective concept. It can be seen as a manner to quantify how much individuals inside the group tend to agree on their opinions. In a purely collaborative system, the only data we have about users' opinions are their historical evaluations over the set of items. Therefore, these evaluations were the base to judge each group's homogeneity degree.

6.4.2 Obtaining a dissimilarity matrix

The first step for the definition of groups was the creation of a dissimilarity matrix over the users. That is, a matrix $\boldsymbol{D}$ of size $n \times n$ (n is the number of individuals) where each D_{ij} contains the dissimilarity value between individuals i and j. To build this matrix it is only necessary to calculate one matrix's diagonal, as the dissimilarity is symmetric. In order to calculate the similarity between each pair of users, we used the following steps:

1. The correlation coefficient between the two users (ρ_{ij}) is calculated (Equation 2.1). The correlation coefficient is as a similarity value, which varies between -1 (minimum similarity) to +1 (maximum similarity).
2. The previous result is transformed into a dissimilarity value, varying between 0 (smallest dissimilarity) and 1 (maximum dissimilarity). For this transformation we used the simple formula:

$$\text{dissim}(i, j) = 1 - \frac{\rho_{ij} + 1}{2} \quad . \tag{6.1}$$

The dissimilarities between users were thereafter used to construct groups with the three different homogeneity degrees desired, as explained in the next section.

6.4.3 Trying to form groups by controlling the dissimilarity

With the dissimilarity matrix we are able to say for each pair of users if they are similar or not. How can we extend this notion to a group of users?

10 This happens in the real life. For example, one person go to the movies with different groups of friends.

Figure 6.1 shows a histogram of the dissimilarity between pairs of users. It can be noted that it has a distribution approximately normal. Table 6.1 shows the mean, standard deviation and Tukey's five number summary for the variable.

One method to form groups with a pre-defined degree of homogeneity is the following:

1. Define a threshold for the dissimilarity. It is known that in a normally distributed variable, about 68% of individual values lie between *mean ±1 standard deviation*. And approximately 95% of values lie between *mean ±2 standard deviation*. We could, for example, use the latter value.
2. The groups with high homogeneity degree and size k would be formed by k users such that all dissimilarities between them (all the $\frac{k(k-1)}{2}$) are less than the "left" threshold (e.g. mean - 2 standard deviation). Analogously, the groups with low homogeneity degree would be formed by k users such that all dissimilarities between then are larger than the "right" threshold.

The problem with this methodology is its complexity. We can easily express this problem as a problem in graphs. If we consider each user as a vertex in the graph and the dissimilarity between each pair of users (a, b) as the weight of the undirected edge (as the dissimilarity is symmetric) between a and b, we can represent the dissim-

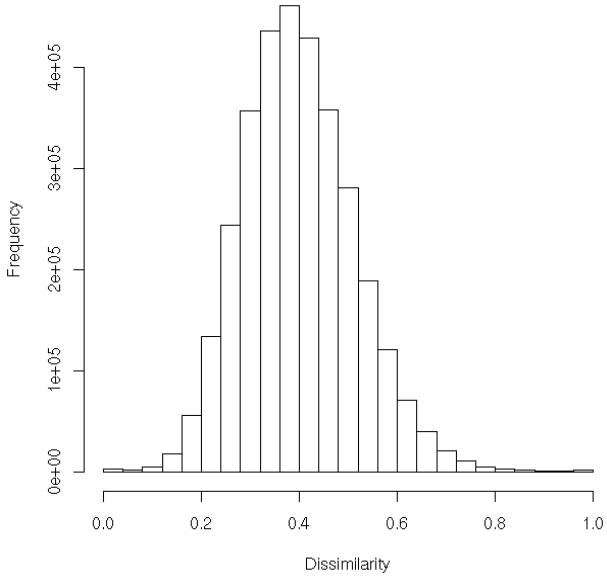


Figure 6.1 Histogram of the dissimilarity between pairs of users

Table 6.1 Tukey's five number summary, mean and standard deviation for the dissimilarity between pairs of users.

	Minimum	*1st Quartile*	*Median*	*3rd Quartile*	*Maximum*	*Mean*	*Standard deviation*
Dissimilarity	0.0000	0.3193	0.3920	0.4719	1.0000	0.3996	0.1150

ilarity matrix as a complete[11], undirected graph with weighted edges. Figure 6.2 shows a hypothetical dissimilarity matrix for the individuals {a, b, c, d} represented as a graph.

Therefore this problem is how to find complete subgraphs (also known as *cliques*) with k vertices, such that all edges in these subgraphs have weights smaller than a threshold (in the case of groups with high homogeneity. The case of groups with low homogeneity is analogous). Figure 6.3 depicts a clique of size 3 (corresponding to a group of 3 individuals with high homogeneity) found in the graph of Figure 6.2, considering a hypothetical threshold of 0.4.

User	***a***	***b***	***c***	***d***
a	-			
b	0.7	-		
c	0.2	0.5	-	
d	0.1	0.1	0.3	-

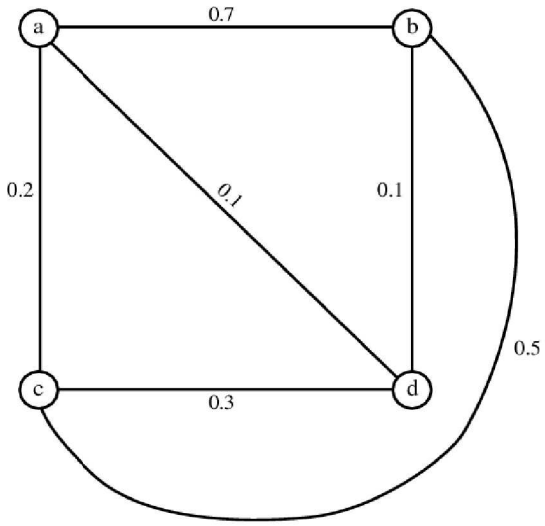


Figure 6.2 Representing a dissimilarity matrix as an undirected complete graph

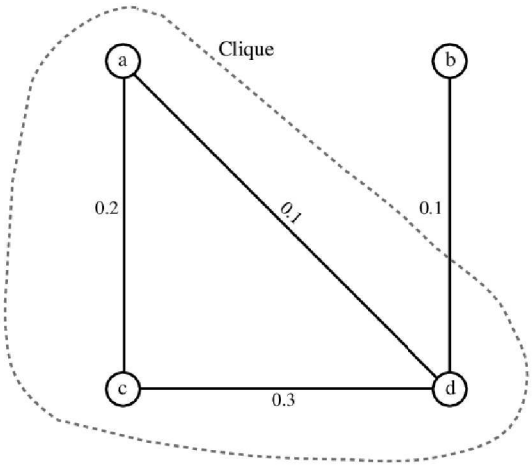


Figure 6.3 Clique of size 3 corresponding to a homogeneous group with threshold 0.4

11 In a complete graph each pair of distinct vertices is joined by exactly one edge. Additionally, each edge joins a pair of distinct vertices [8].

However, the problem "given an undirected graph and an integer k, determine if the graph contains a clique of size $\geq k$" is a known NP-complete problem [38]. Therefore there is no algorithm that can solve our problem in polynomial time, for if it existed we could use it to solve the original clique problem in the following way:

- Use the putative polynomial algorithm to find the k-sized cliques (algorithm that solves our problem).
- If k-sized cliques were found, the answer to the classic clique problem would be affirmative, as if one or more cliques of size k exists it is true that the graph contains a clique of size $\geq k$.
- If k-sized cliques were not found, the answer to the classic clique problem would be negative, as if cliques of size $\geq k$ existed the algorithm would have found (at least one) clique of size k. This would occur because cliques with size $\geq k$ would obligatorily have cliques of size k as subgraphs.

Given the impossibility of using this method efficiently, we adopted a heuristic approach to form the groups. The next section describes it in details.

6.4.4 Forming groups heuristically

Cluster analysis is used to organize a collection of patterns into clusters based on similarity (or dissimilarity). Intuitively, patterns within a cluster are more similar to each other than they are to a pattern belonging to a different cluster [25]. Therefore, clustering can be used to heuristically find groups with high homogeneity (members of the same cluster) and even groups with low homogeneity (members of different clusters).

Forming groups with high homogeneity

Generally speaking, cluster analysis methods are of either of two types [62]:

- **Partitioning methods**: algorithms that divide the dataset into k clusters, where the integer k needs to be specified by the user.
- **Hierarchical methods**: algorithms yielding an entire hierarchy of clusterings of the dataset. *Agglomerative* methods start with the situation where each object in the dataset forms its own little cluster, and then successively merge clusters until only one large cluster remains which is the whole dataset *(bottom-up* approach). Divisive methods start by considering the whole dataset as one cluster, and then split up clusters until each object is separate (*top-down* approach).

Some methods of both types can receive as input a dissimilarity matrix, which is exactly the kind of data we have available.

To form groups with high homogeneity we found more adequate to use a hierarchical method. After all, we needed 100 groups with predefined sizes (3, 6, 12, 24 persons), whereas we would end up with 100 groups of varied sizes if we asked a partitioning method for 100 groups.

Nevertheless, as we had a big dataset (2551 individuals) the clustering process was too expensive (both in time and memory). Furthermore, it was complex to "navigate" into a big hierarchy do find 100 distinct groups, as much homogeneous as possible.

Hence, following the "divide to conquer" philosophy, from the 2551 individuals, we initially selected (with reposition) 100 random groups of 200 individuals each. Then, for each of these groups, we ran the clustering algorithm *divisive analysis* – diana, resulting in 100 different hierarchies. From each hierarchy, the most homogeneous group with 3, 6, 12 and 24 individuals was extracted. In this way all the groups with high homogeneity degree were obtained. We used diana to construct the hierarchies because it is a well-known algorithm first introduced in the classic book of Kaufman and Rousseeuw [29], and a solid implementation for it is available in the statistical package we used, R[12]. Also this algorithm does not need any parameters, which does not bring another source of complexity to our experimental design.

To exemplify how these groups were extracted, suppose we wanted to form a 6-element group from the tree (dendrogram) generated by diana. Figure 6.4 depicts one dendrogram with 20 fictitious objects (remember that during the extraction of groups, each tree has 200 objects). The most homogeneous group with six elements is in the "branch" of the tree with lowest height that has at least six objects. In the figure, the branch with this property is outlined. However, the chosen branch can have a larger number of elements than desired. In the case showed, we wanted a group of six elements, whereas the "best branch" has eight elements. How did we choose the best six elements among these eight candidates? The algorithm we implemented to choose a group of size *n* exhibits the following behavior:

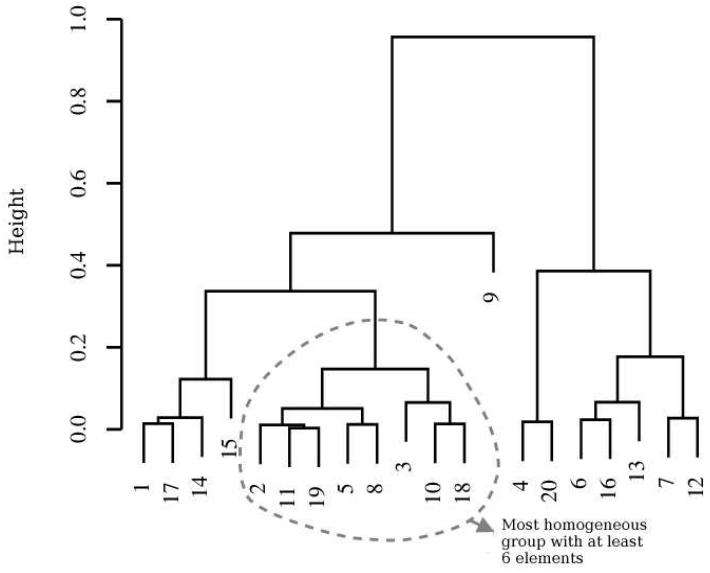


Figure 6.4 Extracting a homogeneous group from a dendrogram

12 R is a free environment for statistical computing and graphics based on the S programming language from Bell Labs. Its official site is http://www.r-project.org. For a quick introduction to R, see [49] (available at the project's site).

1. The "joints" of the groups are scanned from lowest to highest height, until a joint is found such that its branch has at least n (desired) elements.
2. If the branch chosen has exactly n (desired) elements, they form the group. If not, we calculate the total dissimilarity of each combinations of size n from the branch elements (the total dissimilarity is given by summing all $\frac{n(n-1)}{2}$ dissimilarities between the pairs of elements). The group is formed by the combination with smallest total dissimilarity. However, for groups of size 24, it is not possible to test all combinations (e.g. comb(40, 24) > 6.28×10^{10}). For these groups, the following heuristic was applied:
    (a) For each element of the branch, the summation of the dissimilarity between it and all other elements of the branch is calculated.
    (b) The n elements with smallest calculated dissimilarity are chosen.

Forming groups with medium homogeneity

We defined groups with medium homogeneity as those where the dissimilarity behavior is analogous to the general population.

To form a group with medium homogeneity of size k, we randomly selected k users from the population (the 2551 users used in the experiment). In order to avoid surprises due to randomness, after the selection we tested if the average dissimilarity of the obtained group does not differ statistically to the population average. For this, we used a test of comparison between an average (the one from the group) and a specific value (the known population average), with $\alpha = 0.05$ [42]. Using this methodology, we formed all groups with medium homogeneity.

Forming groups with low homogeneity

We employed the same 100 randomly generated groups of 200 individuals each that we used to obtain the groups with high homogeneity in this phase. However, this time we extracted one group with low homogeneity degree (instead of high) for each random group. In order to do this, first we tried a clustering approach:

- First, for each randomly generated group, we used the clustering method *partitioning around medoids* – pam [29], asking for 4 different partitions. This partitions corresponded to the sizes of the desired groups. That is, we used k = 3, 6, 12, 24.
- For each partition (in k clusters) generated for a two hundred sized group, we extracted one k-sized group with low homogeneity, by choosing the most central element from each cluster.

For example, to form a group of 6 persons with low homogeneity, the pam method was run with $k = 6$ for one of the groups with 200 individuals. For each of the 6 clusters generated, the most central element was selected. These elements will form the group we look for. In order to find the most central elements, we calculated for each element the summation of the dissimilarity between it and all others in the same

cluster. The one with smallest summation was considered to be the most central element for each cluster.

Even though the strategy presented looks plausible, it did not work well in practice. Many times the most central elements of the clusters were too near, and the group created was not heterogeneous enough. Probably, given the lack of a well-defined group structure in the data, the groups generated by pam were too close.

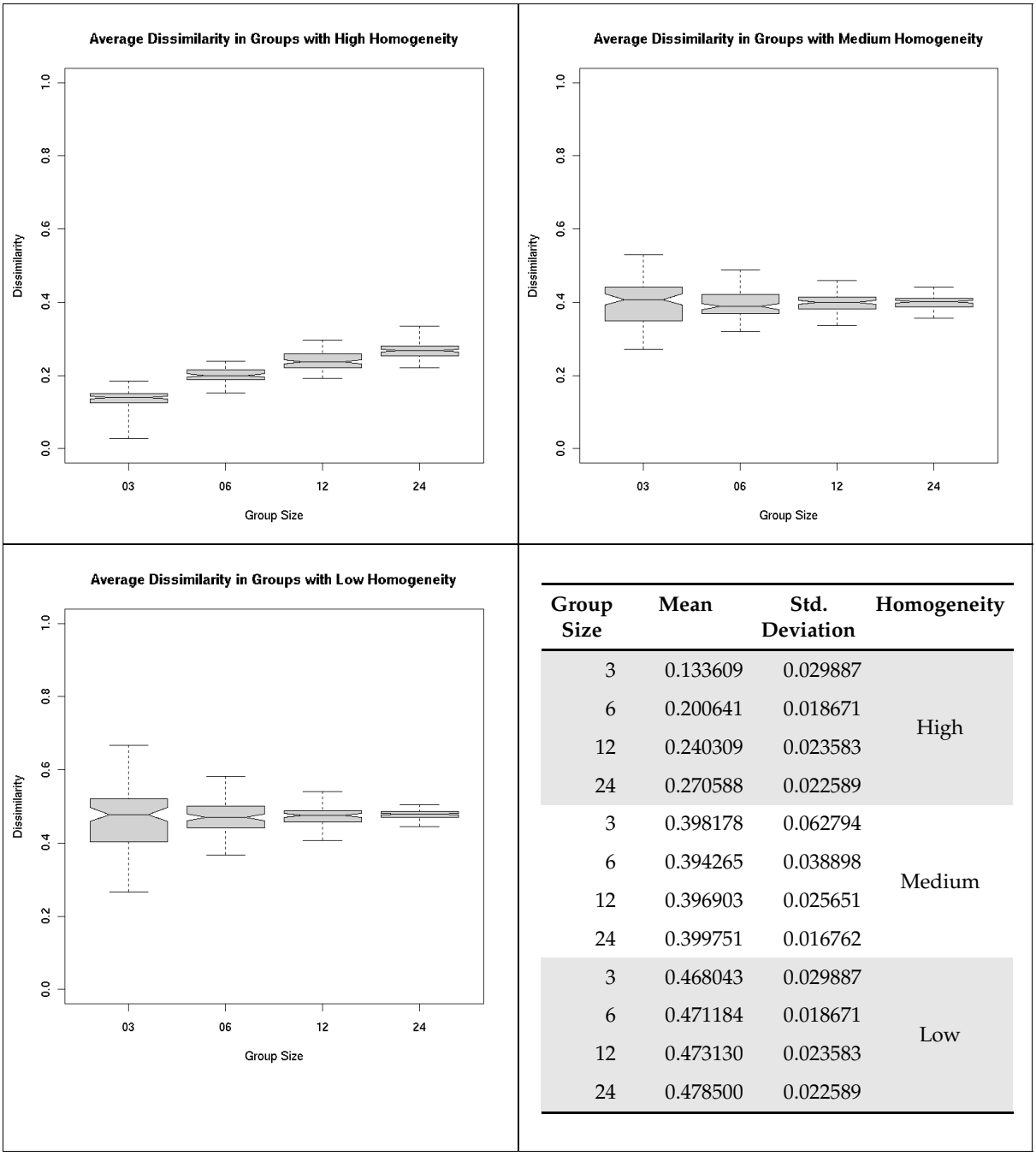
A simpler methodology showed a better performance to form the groups with low homogeneity. For each randomly generated group of 200 individuals, we calculated for each element the summation of the dissimilarity between it and the other 199. The k elements with largest sums were chosen to form a k-sized group. This second methodology was adopted to form the groups.

6.5 Biases in the used data

The criterion to use only individuals with at least 150 evaluations may introduce biases in the data. First, the results encountered may be more effective for active users of the recommendation system. The exclusion of the users with less evaluations also makes the dataset denser and may artificially impact the prediction accuracy. Also, as all users considered are active users, we did not have the situation where a subset of the group that wants a recommendation is very active in the system, while another seldom uses it, consequently having few evaluations.

Another bias is introduced by the process employed to form the groups. As we have formed the groups heuristically based on real data, we do not have a fine control on the homogeneity degree of the groups (measured as the mean dissimilarity between pairs of group members). Figure 6.5 depicts the mean dissimilarity of the groups formed. A bias that can be noted is that there are undesirable variations on the dispersion of the mean dissimilarity inside the same homogeneity level.

Although it would be ideal to do a second phase of experiments on-line to avoid the biases of the off-line data, we believe the results of the off-line experiments are still valid in the real world. Given that the *Eachmovie* dataset is an extensive collection of data from a real recommender system, and various experiments have been conducted using only its data (see e.g. [6], [35], [11]). The fact that all recommendation strategies were run under the same biases and apparently none of them favors one strategy over the others strengthens our assumption that the results were not significantly distorted. This possible deficiencies will be regarded when we analyze the results in Chapter 7.



Group Size	Mean	Std. Deviation	Homogeneity
3	0.133609	0.029887	High
6	0.200641	0.018671	
12	0.240309	0.023583	
24	0.270588	0.022589	
3	0.398178	0.062794	Medium
6	0.394265	0.038898	
12	0.396903	0.025651	
24	0.399751	0.016762	
3	0.468043	0.029887	Low
6	0.471184	0.018671	
12	0.473130	0.023583	
24	0.478500	0.022589	

Figure 6.5 Box plots showing the mean dissimilarity for each type of group (different sizes and homogeneity degrees). Each box plot is generated from the average dissimilarity of the 100 groups of the specified size and homogeneity degree. As usual, the uppermost and lowermost lines are drawn at the highest and lowest values; whereas the three lines that form the box are drawn 25% (first quartile), 50% (median) and 75% (third quartile) of the way through the data. If the notches of two plots do not overlap then the medians are significantly different at the 5 percent level. The table shows the mean and standard deviation for the average dissimilarity of each type of group.

6.6 Evaluation of recommender systems

6.6.1 Defining metrics to evaluate recommender systems

Although standard data sets for testing recommender systems exist (of which *Eachmovie* is one of the most populars), there has been no standardized way to evaluate these systems. As it was pointed out by Herlocker [19], this leads to the impossibility of directly comparing the methods proposed by different researchers, leaves the burden of investigating the best methodologies for evaluating the systems to every researcher and still raises suspiciousness that each researcher has chosen the methodology that gives the best results for his/her method.

In order to cope with this problem, Herlocker—working in the context of recommender systems for individuals—identified six tasks that can be performed by the users of an information-filtering system and analyzed the suitability of ten metrics in view of those tasks.

The six tasks as identified by Herlocker in [19], were:

1. A user wants to locate a single item whose value exceeds a threshold. For example, a common task would be to locate a single decent movie to watch, or a book to read next.

2. A user is about to make a selection decision that has significant cost, and wants to know what the best option is. For example, a selection between many different health plans (HMOs) could have significant future consequences on a person. They are going to want to make the best possible selection.

3. A user has a fixed amount of time or resources, and wants to see as many of the most valuable items as possible within that restriction. Therefore, the user will be interested in the top n items, where n depends of the amount of time the user has. For example, consider news articles. People generally have a fixed amount of time to spend reading news (such as a half-hour before starting work). In that time, they would like to see the news articles that are most likely to be interesting.

4. A user wants to gain or maintain awareness within a specific content area. Awareness in this context means knowing about all relevant events or all events above a given level of interest to the user. For example, a person in public relations for a company might want to be sure to read all articles that might have an effect on the stock price of the company.

5. A user wants to examine a stream of information in a given order, consuming items of value and skipping over items that are not interesting or valuable. For example, in Usenet bulletin boards, some readers frequently examine the subject line of every article posted to a group. If a subject appears interesting, the entire article is retrieved and read.

6. A user has a single item and wants to know if the item is worth consuming. For example, an user may see an advertisement for a new book and wants to know if it is worth reading.

These tasks demand two different sets of features from a recommender system:

1. Tasks 1-4 are *ranking-related (top-N recommendation problem)*. A recommender system must be capable of ranking collections of items to implement these tasks.
2. Tasks five and six require that the system is capable of giving an absolute value for any arbitrarily chosen item (*prediction problem*). For example, the user could choose a specific book and asks the system how much it "thinks" s/he would like this book.

Given these two categories of problems, Herlocker noticed that we can evaluate recommender systems using metrics of two families: ranking metrics for the *top-N recommendation problem* (generally based on some ranking correlation coefficient); and absolute error metrics for the *prediction problem* (like the mean absolute error).

After empirically evaluating 10 distinct metrics, Herlocker concluded that within each of the two families of metrics there was strong agreement between them. Even disagreements that occurred between the two families of metrics were small and for practical purposes, the choice of evaluation metric did not affect the reported results significantly. Thereupon, it was recommended that the research community should standardize on one or two metrics. It was recommended the mean absolute error because of its simplicity and extensive literature available. For the ranking-based metrics it was not possible to identify one clearly superior, for this matter the choice of one of the ranking-based metrics was not considered clear.

6.6.2 Evaluating recommender systems for groups

Difficulties to evaluate a recommender system for groups

When we introduce the problem of making recommendations for groups, evaluating the recommender system poses a new set of difficulties. Before, when we had recommendations for a single user, it was simple to evaluate the satisfaction of an user given a recommendation in off-line experiments using historical data: to do this, we utilized a fraction of the historical dataset as a test set, made the predictions for items in this set, and then compared the predictions with the real preferences, expressed by the grades given by the user to these items. The final metric is the average of the chosen evaluation metric over all users considered for the experiment. This methodology has been used frequently in the literature (see e.g. [57], [6], [9], [20], [35], [11]).

However, when we consider a group of users, how to measure the group's satisfaction for a given recommendation is a difficult endeavor. The impossibility of having an absolute criterion to determine the level of satisfaction in a group (as pointed out in Chapter 3) implies that there is no metric that can universally quantify the satisfaction of a group. Therefore no metric will be completely "unbiased". For example, a metric that uses the average individual satisfaction implicitly assumes that to have everyone in a group satisfied "on the average" is a good thing (no matter of which metric was used to quantify the individual satisfaction). Psychology research has identified that people's decisions depend not only on their personal satisfaction, but

also on the probable acceptance that their decisions will have among others in the group [39]. While this supports the notion that a "majority-inspired" metric (like the average) is a good one, it has also been noted that in mixed-motive groups this kind of decision leads to global compromise rather than integration of interests. The existence of individuals highly unsatisfied with the group's decisions compromises its long term existence, as these individuals may distance themselves from the group [41]. Therefore, a metric that favors unanimity over majority is also useful.

Characterization of the evaluation problem

Due to the fact that the running systems and filtering algorithms in the literature (that make recommendations for individual users) deal mostly with the *top-N recommendation problem* (e.g. [22], [57], [64], [56], [58], [31], [28], [47]), and also to limit the scope of this work, we focused only in this problem, not trying to evaluate the recommendation strategies in view of the *prediction problem.* Thence, we characterize our problem of evaluating a recommendation for a group as following:

- Given partial individual preference rankings R_1, ..., R_m for the m members of a group;
- and a total order R for the N ranked items.
- Measure the level of satisfaction of the group to R.

The individual preference rankings are built from the actual grades that each user gave to the items from the test set. They are partial orders[13], normally there are ties (items that received the same grade) and not every one of the N items can be compared—the individual may not have evaluated it.

Choosing adequate evaluation metrics

A good metric[14] to quantify the group satisfaction would be a ranking distance proper for the evaluation of recommender systems and at the same time with good "social" characteristics. That is, it should also be as fair as possible to the members of the group.

The Kendall tau distance [30] counts the number of pairwise disagreements between two rankings. That is, given two rankings α and φ, we can see the Kendall's distance K as:

- $K(\alpha, \varphi)$ = number of pairs (i, j) such that $\alpha(i) < \alpha(j)$ but $\varphi(i) > \varphi(j)$ or $\alpha(i) > \alpha(j)$ but $\varphi(i) < \varphi(j)$, where $\alpha(i)$ means the position of i in the ranking α.

13 A partial order is a reflexive ($s \leq s$ for every element s), antisymmetric ($s \leq t$ and $t \leq s$ imply $s = t$) and transitive ($s \leq t$ and $t \leq u$ imply $s \leq u$) relation. [50]

14 It is worth to underline that in this work we use the term "metric" liberally, to mean *a measurement criterion*. That is, the measures used do not necessarily satisfy the mathematical criteria to be considered *metrics*. In fact, the measure we used (Kendall's tau) is not a real metric but it defines a meaningful measure between rankings. For a good characterization of measures to compare lists, including the tau, see [17].

The Kendall tau distance normalized to assume values between -1 (total disagreement—reverse rankings) and +1 (complete agreement—identical rankings) is the Kendall's ranking correlation coefficient, denoted by the Greek letter τ (tau).

In our case, we want to measure the distance of several partial rankings (individual preferences) to one full ranking (the group recommendation). The τ can be generalized to accommodate this need. One obvious way to do it is to take the average of the τ's between each individual preference ranking and the group recommendation (this generalization is used in [16]). That is

$$\tau_{avg} = \frac{1}{m} \sum_{i=1}^{m} \tau_{|R_i}(R_i, R) \quad \text{, where:} \tag{6.2}$$

- R_i, $i \in \{1, ..., m\}$ are the individual rankings for the m group members and R is the group recommendation;
- $\tau_{|R_i}(R_i, R)$ is the Kendall's ranking correlation coefficient between R_i and R, restricting the rankings R_i and R to the elements contained in $R_i \cap R$.

The average τ_{avg} has a good social characteristic. One ranking R with largest τ_{avg} is a *Kemeny optimal aggregation* (it is not necessarily unique). Kemeny optimal aggregations are the only ones that fulfill at the same time the principles of neutrality and consistency of the social choice literature and the *extended Condorcet criterion* [16]:

- If a majority of the individuals prefer *a* to *b*, then *a* should have a higher ranking than *b* in the aggregation.

Kemeny optimal aggregations are NP-hard to obtain when the number of rankings to aggregate is ≥ 4 [16]. In this way it is not possible to implement a strategy that is optimal in view of the average tau, making it a good reference for comparison in practice.

Kendall's τ was one of the measures evaluated by Herlocker [19], and it demonstrated agreement with the other measures when tested empirically. In his theoretical analysis of the measures, he pointed out as one deficiency of the ranking correlation coefficients the fact that they cannot take into account ties. That is, items that are of the same importance to the individual also have their relative order in the ranking considered. For example, if the items *a* and *b* have received the same evaluation, it is indifferent to the user if these items appear in the final ranking as (*a*, *b*) or (*b*, *a*) but the coefficient will penalize one of the two orders, considering it "wrong". We modified the calculation of the τ to consider ties. In this way this spurious penalization does not occur.

Given the good characteristics of the average tau, it was chosen as our main evaluation metric. However, as it was said previously, in some social contexts other factors could be of major importance, like nobody in the group was too dissatisfied. To observe the behavior of the maximum and minimum user's satisfaction, we also observed the maximum and minimum tau, defined as:

$$\tau_{max} = \max_{i=1}^{m} (\tau_{|R_i}(R_i, R)) \quad \text{, and} \tag{6.3}$$

$$\tau_{min} = \min_{i=1}^{m} (\tau_{|R_i}(R_i, R)) \quad . \tag{6.4}$$

***Calculating the* τ**

In order to calculate the τ (without considering ties) between two rankings, we can proceed in the following way:

- Let n be the number of objects in the ranking (its size). The number of pairs of objects AB is therefore $\binom{n}{2} = \frac{n(n-1)}{2}$.
- Let P be the number of agreements between the rankings, initially $P = 0$.
- Let Q be the number of disagreements between the rankings, initially $Q = 0$.
- For each pair AB, do:
  - if both rankings agree with the relative order of AB (i.e., if in both rankings A has a higher rank than B or vice-versa), add 1 to P.
  - otherwise, the rankings disagree with the relative order of AB, consequently add 1 to Q.
- The obtained score S between the two rankings is defined as $S = P - Q$.
- τ is defined as $\frac{Obtained\ score}{Maximum\ possible\ score}$.

The maximum score occurs when the rankings agree in all pairs, that is, $P = n(n-1)/2$, $Q = 0 \Rightarrow$ $S_{max} = n(n-1)/2$. Therefore, we have

$$\tau = \frac{S}{n(n-1)/2} \tag{6.5}$$

As $S = P - Q$ and $P + Q = n(n-1)/2$ we can also express τ as:

$$\tau = \frac{2P}{n(n-1)/2} - 1 \quad \text{, or} \tag{6.6}$$

$$\tau = 1 - \frac{2Q}{n(n-1)/2} \quad . \tag{6.7}$$

In our case, we want to compare how the ranking of the recommendation compares against the user's preferences, which is a ranking with ties. For this reason we

modified Equation 6.6 to take account of ties (this type of modification is proposed in Kendall's original work on ranking correlation methods [30], although the recommender system literature apparently has not made use of it when using the τ to evaluate recommender systems):

- When examining the pairs, we add 1 to P only if the pair *AB* currently being examined is in the same order as the user's preferences and it is not tied. Let's call *P* calculated in this way as *P'*
- Each tie of length *l* in the user's preferences subtract $l(l-1)/2$ in the maximum score possible. Therefore we subtract $\frac{1}{2}\sum_l l(l-1)$ of the maximum score. Consequently, the modified formula used to calculate the τ was

$$\tau = \frac{2P'}{\frac{n(n-1)}{2} - \frac{1}{2}\sum_l l(l-1)} - 1 \tag{6.8}$$

6.7 Applying the evaluation methodology

As seen on Section 6.4, we have data for 4 different group sizes (3, 6, 12 and 24 people) and 3 levels of homogeneity (high, medium and low). For each combination "group size" × "homogeneity degree" we generated 100 repetitions.

In this way, in the experimental design we have 3 factors: group size, homogeneity degree, and the strategy used to generate the recommendation.

We observed three variables, τ_{avg} (Equation 6.2), τ_{max} (Equation 6.3) and τ_{min} (Equation 6.4). The correlation coefficient in each of these formulas was calculated using Equation 6.8.

To sum up, for each combination of "group size" × "homogeneity degree" × "strategy" we have 100 triplets (τ_{avg}, τ_{max}, τ_{min}). We then compare the average of each of these variables (over the 100 repetitions) using a three-way (as we have 3 factors) analysis of variance (ANOVA). That is, in the end we see if there were differences in the behavior of the means: $\overline{\tau}_{avg}$, $\overline{\tau}_{max}$ and $\overline{\tau}_{min}$, given the different levels of each factor. Figure 6.6 depicts the evaluation procedure for a given recommendation strategy.

Each simulation ranked 50 movies from the test set (if the simulation is used to adjust parameters, we use the training set instead, see Section 6.3). Two distinct ways were used to chose these movies for each group. In the first, the 50 movies were chosen randomly from the test set (training set if adjusting parameters); whereas in the second the 50 most seen movies by the group members were chosen from the test set (training set if adjusting parameters). Both methods may have their advantages and disadvantages. When we choose randomly we are not introducing bias by choosing a specific type of movie, however we can choose movies that were rarely seen by the individuals in the group. If this happens, all the τ will be calculated based on a

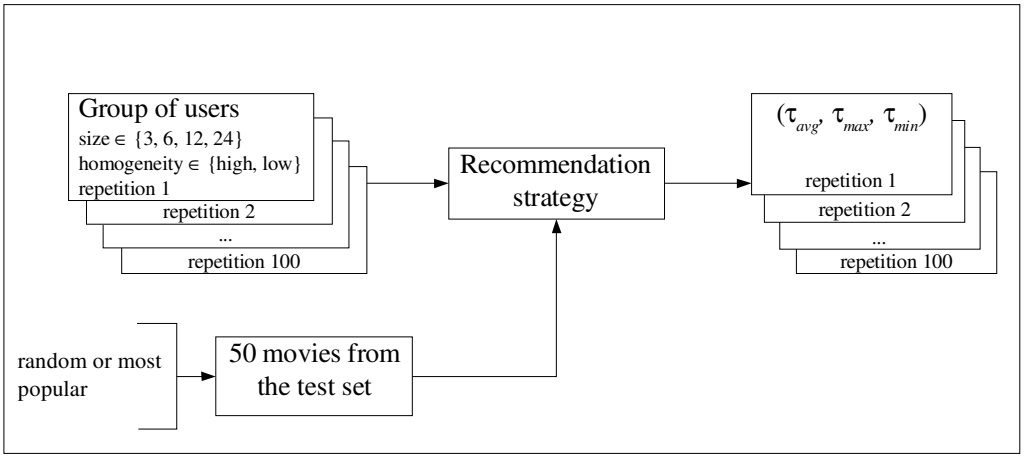


Figure 6.6 Summary of the evaluation process. For each group type (size and homogeneity degree), a recommendation is generated for each repetition. These recommendations are made by ranking 50 movies from the test set. The τ_{avg}, τ_{max} and τ_{min} are calculated for each recommendation. Afterward the averages will be compared using analyses of variance.

small collection of objects (the intersection between the movies ranked and those previously evaluated by each user—see the definition of Equation 6.2). On the other hand, if we choose the most seen movies, the intersection between them and each group member's evaluated movies will likely be large (therefore the τ will be calculated based on a larger number of objects), but choosing movies in this way we can introduce bias (basing the evaluation on the "most popular" movies for the groups).

6.8 Graphical visualization of the results

Graphics are powerful and largely used tools for quickly visualizing data. In the area of computer performance evaluation, a popular choice for quickly visualizing the performance of a system is to use Kiviat graphs. A Kiviat graph is a circular graph in which many performance measures are plotted along radial axes. In its more usual incarnation, an even number of metrics are used. Half of these metrics are "higher-is-better" (HB) metrics, whereas the other half are "lower-is-better" (LB) ones. The HB and LB metrics are plotted along alternate radial axes in the graph. In an ideal system, all HB metrics would be high and all LB metrics would be low. In this case, we would have an ideal Kiviat graph: a star [26].

We will bring Kiviat graphs to our domain, in order to visually describe the performance of the recommendation strategies we evaluated. We utilized three HB and three LB metrics. Our HB metrics were the $\overline{\tau}_{avg}$, $\overline{\tau}_{max}$ and $\overline{\tau}_{min}$. As LB metrics we utilized the standard deviation of the HB metrics. Figure 6.7 shows the measurements for two hypothetical strategies: strategy "Foo" (nearly perfect) and strategy "Bar" (with a good $\overline{\tau}_{avg}$, but a bad $\overline{\tau}_{min}$ and larger standard deviations).

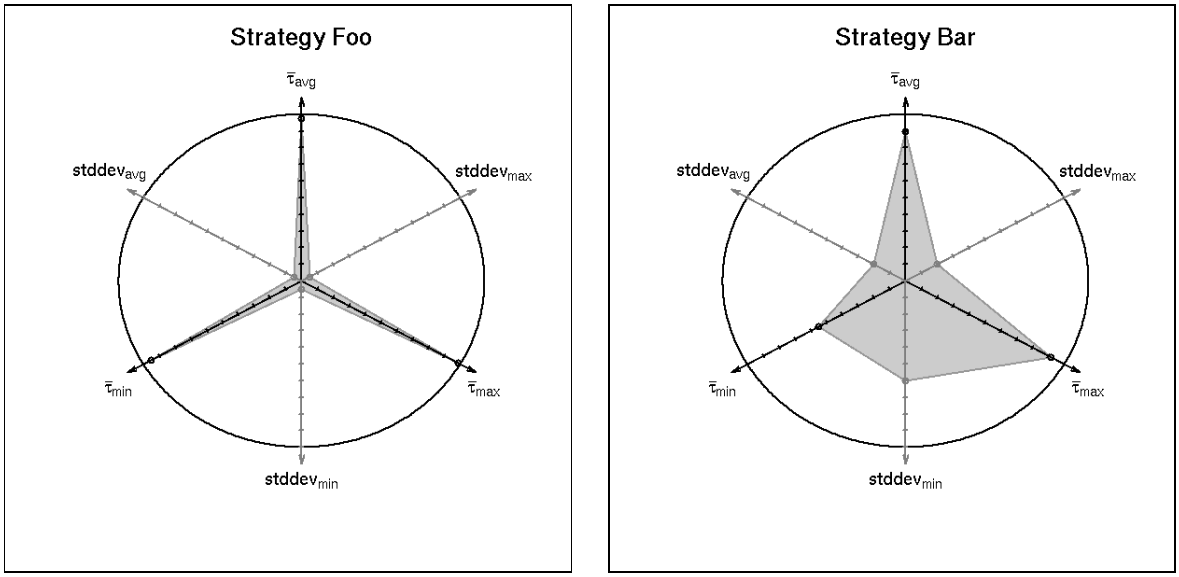


Figure 6.7 Kiviat graphs for the hypothetical group recommendation strategies “Foo” and “Bar”. The Kiviat graph of Foo shows a near-perfect behavior, whereas Bar is clearly inferior based on the metrics chosen.

Chapter 7

Results and Discussion

In this chapter, the recommendation methodologies presented in Chapters 4 and 5 are evaluated using the framework developed in Chapter 6

7.1 Overview

In this chapter, we apply the evaluation methodology described in Chapter 6 to compare the fuzzy technique presented in Chapter 4 to the symbolic model-based technique developed in Chapter 5.

In Section 6.7, we have described two distinct policies to select movies from the test set: a "most selection", which for each group selects the 50 most seen movies (by this group) from the test set; and a "random selection", which selects 50 random movies. We highlighted the pros and cons of each policy. Before starting the experiments, we will analyze if the two different selection policies would lead two different conclusions or if they are equivalent. Then we will introduce the experiments using the fuzzy strategies in Section 7.2 and the experiments using the model-based strategies in Section 7.3. Section 7.4 finishes the chapter by comparing the recommendation strategies.

7.1.1 Defining a selection policy for testing

To investigate if the two selection policies would really give different results, we evaluated the nine configurations considered for the fuzzy method (3 aggregation operators × 3 exploitation operators) under the two options. Then we compared the results.

A run of each configuration generates 12 triplets ($\overline{\tau}_{avg}, \overline{\tau}_{min}, \overline{\tau}_{max}$): 4 group sizes × 3 homogeneity (notice that each tau is averaged over 100 repetitions). Therefore, the running of nine configurations generates 108 triplets ($\overline{\tau}_{avg}, \overline{\tau}_{min}, \overline{\tau}_{max}$). For each tau, we did a *Pearson's product-moment correlation comparison*[15] between the 108 results obtained using "most selection" and the 108 results obtained using "random selection" with $\alpha = 0.99$. Table 7.1 shows the results.

Table 7.1 Tests of correlation using Pearson's product-moment correlation comparison between the average taus obtained when using "most selection" versus the ones obtained using "random selection". Alternative hypothesis: "true correlation is not equal to 0".

metric	*p-value*	*estimated correlation*	*99% confidence interval*	
			lower	upper
$\overline{\tau}_{avg}$	$< 2.2 \times 10^{-16}$	0.998605	0.997695	0.999156
$\overline{\tau}_{min}$	$< 2.2 \times 10^{-16}$	0.963112	0.939740	0.977524
$\overline{\tau}_{max}$	$< 2.2 \times 10^{-16}$	0.971595	0.953470	0.982722

As can it can be seen, the observed means have very high correlations when computed using "most selection" and "random selection". Therefore, we will suppose

15 We did a correlation test instead of a test of difference (like a t-test) because we are not interested if the absolute values are equal, they only need to have a strong correlation to be equivalent for us. For example, if using one selection criteria we obtained the values (x_1, x_2, ..., x_{108}) and using the other we had ($1.1x_1$, $1.1x_2$, ..., $1.1x_{108}$) it's clear that we can discard one of the selection criteria and do all the evaluations using only the other.

that the selection bias envisaged in Section 6.7 is not very strong, and both methods give equivalent results. From now on, we will do the evaluations only under "most selection".

7.2 Experiments using the fuzzy aggregation-based strategies

For each one of the 1200 groups available (4 sizes × 3 homogeneity degrees × 100 repetitions, see Section 6.4), the fuzzy aggregation-based methodology was run, using different quantifiers. We tried nine combinations of quantifiers: {*as many as possible*, *most*, *at least half*} in the aggregation phase × {*as many as possible*, *most*, *at least half*} in the exploitation phase. Only the simpler configuration of the fuzzy majority was tried, the one that considers that every individual has the same importance (see Sections 4.3.1 and 4.4.1).

The goal is to evaluate how the metrics $\overline{\tau}_{avg}$, $\overline{\tau}_{max}$ and $\overline{\tau}_{min}$ are affected by the variation on the size and homogeneity degree of the groups as well as the strategy (quantifiers) used. For this, we use analysis of variance (ANOVA). As we have three factors (size, homogeneity degree and strategy), we will use three-factor analysis of variance (three-way ANOVA). For each observed metric ($\overline{\tau}_{avg}$, $\overline{\tau}_{max}$ and $\overline{\tau}_{min}$) one univariate analysis of variance was performed. Tables 7.2, 7.3 and 7.4 show the analysis of variance tables.

Table 7.2 Analysis of variance for the metric $\overline{\tau}_{avg}$

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
strategy	8	0.053	0.007	0.8350	0.5716	
groupSize	3	0.669	0.223	28.2712	<2e-16	***
homogeneity	2	280.642	140.321	17778.6026	<2e-16	***
strategy:groupSize	24	0.008	0.0003324	0.0421	1.0000	
strategy:homogeneity	16	0.056	0.003	0.4422	0.9718	
groupSize:homogeneity	6	5.531	0.922	116.7964	<2e-16	***
strategy:groupSize:homogeneity	48	0.015	0.0003132	0.0397	1.0000	
Residuals	10692	84.389	0.008			

Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Table 7.3 Analysis of variance for the metric $\overline{\tau}_{min}$

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
strategy	8	0.03	0.004176	0.1437	0.9971	
groupSize	3	116.79	38.93	1339.6922	<2e-16	***
homogeneity	2	522.54	261.27	8991.2452	<2e-16	***
strategy:groupSize	24	0.03	0.001240	0.0427	1.0000	
strategy:homogeneity	16	0.04	0.002447	0.0842	1.0000	
groupSize:homogeneity	6	8.83	1.47	50.6214	<2e-16	***
strategy:groupSize:homogeneity	48	0.07	0.001480	0.0509	1.0000	
Residuals	10692	310.69	0.03			

Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Table 7.4 Analysis of variance for the metric $\overline{\tau}_{max}$

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
strategy	8	0.068	0.008	0.5896	0.7873	
groupSize	3	77.011	25.670	1781.8610	<2e-16	***
homogeneity	2	81.643	40.821	2833.5456	<2e-16	***
strategy:groupSize	24	0.029	0.001	0.0847	1.0000	
strategy:homogeneity	16	0.057	0.004	0.2486	0.9989	
groupSize:homogeneity	6	19.688	3.281	227.7688	<2e-16	***
strategy:groupSize:homogeneity	48	0.056	0.001	0.0805	1.0000	
Residuals	10692	154.034	0.014			

Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Tables 7.2, 7.3 and 7.4 show that size of the group and the homogeneity degree have statistically significant influence in the three observed metrics. However, no significant influence was observed for the strategy. Also, no interaction effects involving the strategy was observed. Consequently, the behavior of the nine quantifier configurations tested was equivalent.

It can also be observed the interaction effects (for the three observed metrics) between the group size and the homogeneity degree. This may be attributed to one bias in the data discussed in Section 6.5: under the same homogeneity degree, groups of different sizes have different dispersion of the average dissimilarity (see Figure 6.5).

As different strategies showed no significant difference, we will take just one of them to carry on the comparisons with model based methodologies (Section 7.4). We will choose the strategy "As Many As Possible + Most", as it reached the highest (but not significantly different) value for $\overline{\tau}_{avg}$ and $\overline{\tau}_{min}$, and the second highest for $\overline{\tau}_{max}$. Table 7.5 shows the (global) means by strategy.

Table 7.5 Grand means by strategy. The shaded cell of each column corresponds to the highest value observed for the metric

Strategy	$\overline{\tau}_{avg}$	$\overline{\tau}_{min}$	$\overline{\tau}_{max}$
As Many As Possible + Most	0.400735	0.079524	0.670653
Most + Most	0.400411	0.076250	0.671371
As Many As Possible + As Many As Possible	0.399357	0.078834	0.669051
Most + As Many As Possible	0.399341	0.077396	0.670607
At Least Half + As Many As Possible	0.398514	0.078040	0.668095
As Many As Possible + At Least Half	0.398502	0.078360	0.665972
At Least Half + Most	0.398047	0.076511	0.669070
Most + At Least Half	0.397807	0.075240	0.668112
At Least Half + At Least Half	0.392731	0.073654	0.662840

We will defer the comparison of the observed $\overline{\tau}_{avg}$, $\overline{\tau}_{min}$ and $\overline{\tau}_{max}$ under different group sizes and homogeneity degrees until Section 7.4, where we will include model-based results in the comparison.

7.3 Experiments using the model-based strategies

The model-based recommendation technique we developed in Chapter 5 has many parameters that can be adjusted to try to tune its performance. Table 7.6 summarizes the parameters considered (for an explanation of their meaning, see Chapter 5).

To adjust these parameters, we run different configurations and observed the $\overline{\tau}_{avg}$, $\overline{\tau}_{min}$ and $\overline{\tau}_{max}$ for each of them. However, to do this adjustment, we could not use the test set, because test set data must be used only to verify the final performance of models, not to adjust them. For this, we used a *training set*, as mentioned in Section 6.3.

Each configuration investigated was run for the 1200 groups available. For each group, the configuration ranked the 50 most seen movies from the training set by the group (see Sections 6.7 and 7.1.1 for the reasoning of this selection policy). We then observed the global means by configuration for the metrics $\overline{\tau}_{avg}$, $\overline{\tau}_{min}$ and $\overline{\tau}_{max}$.

However, as can be seen in Table 7.6, testing all possible configurations implies 1440 runs, too costly for us to execute. Therefore, we adopted a simple "greedy" approach (that can lead to suboptimal configurations): starting from a simple configuration, we selected one parameter at each step, in the following way (each subsequent step starts from the configurations that so far achieved the best results for each metric):

- the starting configuration had 30 users, used the similarity measure based on Euclidean distance, no weights, no deformation, and did not use the number of common items to weight the comparisons;
- in the first step, we varied the number of users in the item prototype. The best results (for all three metrics) was achieved by the configuration with 300 users.
- in the second step we varied the similarity measure. The weighted affinity surpassed the metric based on Euclidean distance for all three metrics.
- the third step tested the different weights w_j. The best result for the $\overline{\tau}_{avg}$ was no weights, for the $\overline{\tau}_{min}$ was the maximum between the fraction that evaluated the item in each prototype (maxfreq), and for the $\overline{\tau}_{max}$ was the entropy of the group prototype (entropgroup).
- in the fourth step we tested the deformations. They did not improve the results of any metric.
- in the fifth and last step, weighting using the maximum intersection achieved the best results for the $\overline{\tau}_{avg}$ and $\overline{\tau}_{max}$, and weighting using the median achieved the best result for the $\overline{\tau}_{min}$.

In this way, the best configuration obtained for the $\overline{\tau}_{avg}$ was using 300 users, weighted affinity coefficient, no weights w_j, no deformation, and final weighting by the maximum intersection size. For the $\overline{\tau}_{min}$ the best configuration was with 300

users, weighted affinity coefficient, maxfreq as weight when comparing the histograms, no deformation and final weighting by the median intersection size. Whereas for the $\overline{\tau}_{max}$ the best configuration was with 300 users, weighted affinity coefficient, entropy in the histogram of the group prototype as weight, no deformation and final weighting by the maximum intersection size. These three configurations (showed in Table 7.7) will be the ones compared with the results obtained using aggregation methods, in Section 7.4.

Table 7.8 shows the $\overline{\tau}_{avg}$, $\overline{\tau}_{min}$ and $\overline{\tau}_{max}$ (the global means by configuration) for every tested configuration.

Table 7.6 Parameters considered for the group-model methodology. Considering all combinations of values, we would have 1440 possible configurations.

Parameter	*Considered values*	*Number of values*
number of users in the item prototype	30, 50, 100, 200, 300	5
similarity measures	based on Euclidean distance, based on correlation, weighted affinity	3
weights w_j	no weights, maximum between fractions, minimum between fractions, entropy in the group prototype, standard deviation in the group prototype, entropy in the item prototype, standard deviation in the item prototype, similarity between items	8
histogram deformations	none, {3, 2, 1, 1, 2, 3}, {3, 3, 2, 1, 2, 3}	3
weight by the number of common items	no, using mean intersection, using maximum intersection, using median intersection	4
	Number of possible configurations	1440

Table 7.7 Parameters used in the three selected configurations of the symbolic model.

Name	*#Users*	*Similarity Measure*	*Weight w_j*	*Deform.*	*Final adjust*
Symbolic 1	300	affinity	group entropy	none	size max. intersection
Symbolic 2	300	affinity	maximum frequency	none	size median intersection
Symbolic 3	300	affinity	none	none	size max. intersection

Table 7.8 Configurations tested for the various model parameters*.

$\overline{\tau}_{avg}$	$\overline{\tau}_{min}$	$\overline{\tau}_{max}$	# Users	Similarity measure	Weight w_j	Deformation	Final adjust
0.377719	0.070882	0.660045	*30*	euclidean	none	none	none
0.387514	0.081215	0.667194	*50*	euclidean	none	none	none
0.381909	0.082282	0.663964	*100*	euclidean	none	none	none
0.396039	0.088804	0.678896	*200*	euclidean	none	none	none
0.403835	0.098829	0.683450	*300*	euclidean	none	none	none
0.407697	0.101898	0.688473	300	*affinity*	none	none	none
0.392052	0.082223	0.668215	300	*correlation*	none	none	none
0.404688	0.096523	0.687232	300	affinity	*stddevitem*	none	none
0.406599	0.102730	0.669937	300	affinity	*maxfreq*	none	none
0.393771	0.082314	0.689055	300	affinity	*entropitem*	none	none
0.401489	0.095783	0.690479	300	affinity	*entropgroup*	none	none
0.402341	0.096290	0.689715	300	affinity	*stddevgroup*	none	none
0.404957	0.099621	0.686345	300	affinity	*simitemitem*	none	none
0.402692	0.099862	0.659189	300	affinity	*minfreq*	none	none
0.403467	0.097341	0.682064	300	affinity	none	*{ 3 3 2 1 2 3 }*	none
0.403295	0.096026	0.682938	300	affinity	none	*{ 3 2 1 1 2 3 }*	none
0.405867	0.099646	0.669539	300	affinity	maxfreq	*{ 3 2 1 1 2 3 }*	none
0.406440	0.100996	0.670656	300	affinity	maxfreq	*{ 3 3 2 1 2 3 }*	none
0.397049	0.090743	0.681767	300	affinity	entropgroup	*{ 3 3 2 1 2 3 }*	none
0.396757	0.089484	0.682562	300	affinity	entropgroup	*{ 3 2 1 1 2 3 }*	none
0.409936	0.102153	0.688730	300	affinity	none	none	*median*
0.409495	0.101748	0.688456	300	affinity	none	none	*mean*
<u>0.410033</u>	0.101721	0.689004	300	affinity	none	none	*max*
0.409029	<u>0.103130</u>	0.671156	300	affinity	maxfreq	none	*median*
0.408707	0.102643	0.670373	300	affinity	maxfreq	none	*mean*
0.409086	0.102159	0.672234	300	affinity	maxfreq	none	*max*
0.401489	0.095783	0.690479	300	affinity	entropgroup	none	*median*
0.403467	0.096019	0.690639	300	affinity	entropgroup	none	*mean*
0.404001	0.095497	<u>0.691446</u>	300	affinity	entropgroup	none	*max*

* The shadings in the table defines each step. In each step the parameter under consideration is in italic. The best value achieved for each metric is underlined. The configurations that achieved these values were selected.

7.4 General comparison of the recommendation strategies

In this section we compare the fuzzy strategy selected to the model-based strategies and a baseline model. We also use a baseline model to see how well the tested strategies perform when compared against a simple model.

In group decision research, in the domain of social psychology, one baseline model that has been used (see e.g. [24]) is the so called "null model", that takes the opinion of one randomly chosen group member as the group decision. That is, it is a kind of "random dictator" decision scheme. Taking this to the domain of recommender systems, we randomly select one group member and make recommendations for this individual (using traditional neighborhood-based collaborative filtering, as in Section 2.3.1). These recommendations are taken as the group recommendations.

First we performed a three-way ANOVA to verify if all considered factors were significant in the behavior of the observed metrics. Tables 7.9, 7.10 and 7.11 show the analysis of variance tables for each of $\overline{\tau}_{avg}$, $\overline{\tau}_{min}$ and $\overline{\tau}_{max}$. As can be seen in these tables, for the three metrics all observed factors were significant.

Table 7.9 Analysis of variance table for the metric $\overline{\tau}_{avg}$

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
strategy	4	0.277	0.069	9.2035	2.070e-07	***
groupSize	3	1.494	0.498	66.0943	< 2.2e-16	***
homogeneity	2	115.150	57.575	7643.6630	< 2.2e-16	***
strategy:groupSize	12	1.813	0.151	20.0614	< 2.2e-16	***
strategy:homogeneity	8	2.351	0.294	39.0142	< 2.2e-16	***
groupSize:homogeneity	6	0.456	0.076	10.0963	3.927e-11	***
strategy:groupSize:homogeneity	24	1.976	0.082	10.9333	< 2.2e-16	***
Residuals	5940	44.742	0.008			
Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1						

Table 7.10 Analysis of variance table for the metric $\overline{\tau}_{min}$

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
strategy	4	1.408	0.352	12.7382	2.525e-10	***
groupSize	3	88.283	29.428	1065.1906	< 2.2e-16	***
homogeneity	2	216.205	108.103	3912.9738	< 2.2e-16	***
strategy:groupSize	12	1.105	0.092	3.3339	7.454e-05	***
strategy:homogeneity	8	3.752	0.469	16.9764	< 2.2e-16	***
groupSize:homogeneity	6	2.407	0.401	14.5227	< 2.2e-16	***
strategy:groupSize:homogeneity	24	2.109	0.088	3.1811	2.604e-07	***
Residuals	5940	164.103	0.028			
Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1						

Table 7.11 Analysis of variance table for the metric $\overline{\tau}_{max}$.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
strategy	4	0.448	0.112	6.8528	1.687e-05	***
groupSize	3	24.548	8.183	500.2835	< 2.2e-16	***
homogeneity	2	24.925	12.463	761.9653	< 2.2e-16	***
strategy:groupSize	12	1.598	0.133	8.1395	2.188e-15	***
strategy:homogeneity	8	2.440	0.305	18.6499	< 2.2e-16	***
groupSize:homogeneity	6	3.534	0.589	36.0135	< 2.2e-16	***
strategy:groupSize:homogeneity	24	1.516	0.063	3.8620	6.356e-10	***
Residuals	5940	97.155	0.016			
Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1						

We proceed by comparing the means obtained for each variable under the different group sizes, homogeneity degrees and strategy. In order to do this, we used *Tukey Honest Significant Differences* test, a test appropriated for comparing multiple levels of a factor in an analysis of variance . We used TukeyHSD at the 95% confidence level.

7.4.1 Comparisons under low homogeneity

Tables 7.12, 7.13 and 7.14 show the means observed for the metrics $\overline{\tau}_{avg}$, $\overline{\tau}_{min}$ and $\overline{\tau}_{max}$, respectively. In this scenario we can see that the symbolic aggregation-based strategies were the clear winners for groups of 3 and 6 people. For groups of 12 persons, all strategies had similar behavior, whereas for groups of 24 persons the fuzzy strategy was the winner for the $\overline{\tau}_{avg}$ and for the $\overline{\tau}_{min}$ and $\overline{\tau}_{max}$ the results for these large groups were not significantly different. At first sight, the fact that the $\overline{\tau}_{avg}$ for groups of 24 persons have been equivalent or better than ones for groups of 12 persons may seen strange, after all it should be more difficult to make a recommendation to please 24 very different persons than to please 12 persons. Notice, however, that when there are 24 persons, even that we are "formally" considering them at the same homogeneity level that the other smaller groups shown, the average dissimilarity of these large groups may be smaller and have a "narrower" dispersion (see Figure 6.5).

The good performance of the model-based strategies under these circumstances suggests that when the persons have very different preferences, trying to integrate them does not lead to a good decision. It may be better to try to model the group to search for a good compromise.

Table 7.12 Means observed for the $\overline{\tau}_{avg}$ in low homogeneity groups. Two values in the same column followed by at least one lowercase letter in common do not differ statistically at the 5% level. Two values in the same row followed by at least one uppercase letter do not differ statistically at the 5% level according to TukeyHSD test[16].

	3		*6*		*12*		*24*	
Fuzzy	0.146118	bC	0.162499	bC	0.198916	aB	0.244075	aA
Null model	0.144814	bB	0.141697	bB	0.177784	aB	0.220032	bA
Symbolic 1	0.315788	aA	0.251668	aB	0.182358	aC	0.169590	dC
Symbolic 2	0.323616	aA	0.250209	aB	0.181440	aC	0.184698	cdC
Symbolic 3	0.317230	aA	0.254016	aB	0.193988	aC	0.194481	cC

Table 7.13 Means observed for the $\overline{\tau}_{min}$ in low homogeneity groups

	3		*6*		*12*		*24*	
Fuzzy	-0.067283	bA	-0.186861	bB	-0.251238	bcBC	-0.278402	aC
Null model	-0.077064	bA	-0.198570	bB	-0.265226	cBC	-0.316984	aC
Symbolic 1	0.117848	aA	-0.055047	aB	-0.195587	abC	-0.303022	aD
Symbolic 2	0.146038	aA	-0.050153	aB	-0.216830	abcC	-0.319815	aD
Symbolic 3	0.124795	aA	-0.051461	aB	-0.180316	aC	-0.288173	aD

Table 7.14 Means observed for the $\overline{\tau}_{max}$ in low homogeneity groups

	3		*6*		*12*		*24*	
Fuzzy	0.358560	bD	0.505805	bcC	0.610553	aB	0.742350	aA
Null model	0.370139	bD	0.487295	cC	0.592235	aB	0.736154	aA
Symbolic 1	0.536097	aB	0.585554	aB	0.593272	aB	0.709920	aA
Symbolic 2	0.514979	aB	0.535763	abcB	0.555711	aB	0.695646	aA
Symbolic 3	0.525492	aC	0.574912	abBC	0.603703	aB	0.715509	aA

7.4.2 Comparisons under medium homogeneity

Tables 7.15, 7.16 and 7.17 show this scenario. Here an interesting trend start to appear: the null model performs well when the group homogeneity is not too low. The null model was not defeated by any other strategy for groups of 3 and 6 people, for all three metrics considered. For the larger groups the fuzzy aggregation-based method has defeated it for the $\overline{\tau}_{avg}$, but not for the $\overline{\tau}_{min}$ and $\overline{\tau}_{max}$. Another tend-

16 This is valid for all tables of means in this chapter. Therefore we will not repeat this instruction.

ency we see is that the model-based methods did not perform well in groups of 12 and 24 persons for the $\overline{\tau}_{avg}$.

Table 7.15 Means observed for the $\overline{\tau}_{avg}$ in medium homogeneity groups

	3		*6*		*12*		*24*	
Fuzzy	0.438996	aA	0.438567	aA	0.439562	aA	0.434938	aA
Null model	0.418464	aA	0.405054	abA	0.411569	bA	0.401624	bA
Symbolic 1	0.420773	aA	0.387148	bB	0.387280	cB	0.364640	cB
Symbolic 2	0.420835	aA	0.386528	bB	0.388090	cB	0.379361	cB
Symbolic 3	0.420896	aA	0.389660	bB	0.391841	bcAB	0.378076	cB

Table 7.16 Means observed for the $\overline{\tau}_{min}$ in medium homogeneity groups

	3		*6*		*12*		*24*	
Fuzzy	0.277806	aA	0.149794	aB	0.051313	aC	-0.094579	aD
Null model	0.251176	aA	0.108636	aB	0.037157	aC	-0.113674	aD
Symbolic 1	0.253269	aA	0.135179	aB	0.038644	aC	-0.106629	aD
Symbolic 2	0.255466	aA	0.137762	aB	0.047280	aC	-0.097540	aD
Symbolic 3	0.254157	aA	0.141323	aB	0.041759	aC	-0.096746	aD

Table 7.17 Means observed for the $\overline{\tau}_{max}$ in medium homogeneity groups

	3		*6*		*12*		*24*	
Fuzzy	0.597886	aD	0.661434	aC	0.726755	aB	0.788787	aA
Null model	0.586334	aD	0.644410	abC	0.707665	abB	0.763802	abA
Symbolic 1	0.577679	aD	0.618651	bcC	0.675367	cB	0.717880	bcA
Symbolic 2	0.571118	aC	0.599150	cC	0.659978	cB	0.714746	cA
Symbolic 3	0.575098	aC	0.611934	bcC	0.676868	bcB	0.729184	bcA

7.4.3 Comparisons under high homogeneity

This scenario is shown by Tables 7.18, 7.19 and 7.20. Here the null model has really excelled. As it was somewhat expected, when we have a group of people with preferences highly homogeneous, knowing the preferences of one is enough to satisfy all. But also, the aggregation-method has not loosed under any circumstances. Alas, to aggregate similar preferences is easy. The model-based strategy, on the other hand, could only keep up with the others for the $\overline{\tau}_{min}$ (where all strategies showed the same behavior), often achieving a lower value for the others variables. This indicates

that it is not a good candidate method for highly homogeneous group. Other fact we observed from the comparisons with all homogeneity degrees is that the 3 symbolic models used showed the same behavior in almost all cases. This suggests that we could adopt the simplest one (Symbolic 3) in favor of the others.

Table 7.18 Means observed for the $\overline{\tau}_{avg}$ in high homogeneity groups

	3		*6*		*12*		*24*	
Fuzzy	0.586874	aA	0.588048	aA	0.575546	aAB	0.554677	aB
Null model	0.574249	aA	0.573302	abA	0.557986	abAB	0.536197	aB
Symbolic 1	0.561547	aA	0.549512	bA	0.536202	bA	0.504595	bB
Symbolic 2	0.561946	aA	0.548709	bA	0.536333	bAB	0.510793	bB
Symbolic 3	0.561112	aA	0.561112	bA	0.539482	bAB	0.510796	bB

Table 7.19 Means observed for the $\overline{\tau}_{min}$ in high homogeneity groups

	3		*6*		*12*		*24*	
Fuzzy	0.454811	aA	0.398671	aB	0.306598	aC	0.193654	aD
Null model	0.436209	aA	0.385154	aA	0.286217	aB	0.168782	aC
Symbolic 1	0.440115	aA	0.370623	aB	0.284749	aC	0.159142	aD
Symbolic 2	0.442399	aA	0.367133	aB	0.279202	aC	0.152963	aD
Symbolic 3	0.437630	aA	0.375957	aB	0.287338	aC	0.158579	aD

Table 7.20 Means observed for the $\overline{\tau}_{max}$ in high homogeneity groups

	3		*6*		*12*		*24*	
Fuzzy	0.705626	aC	0.747102	aB	0.789610	aA	0.813364	aA
Null model	0.696531	aC	0.736897	aB	0.777073	aA	0.804479	aA
Symbolic 1	0.674948	aB	0.699215	bB	0.738829	bA	0.764428	bA
Symbolic 2	0.674048	aC	0.700596	bC	0.730980	bB	0.767908	bA
Symbolic 3	0.674216	aB	0.702084	bB	0.737784	bA	0.770239	bA

7.4.4 The importance of the homogeneity degree

In various cases observed in the last section, there was no difference in the behavior of a recommendation strategy when used for different group sizes (the comparisons by rows in the tables). For the homogeneity degree, however, we observed significant difference in *all* cases, for the metric $\overline{\tau}_{avg}$ (which is the most important of the three). For the other two metrics, significant difference was also observed in most of

the cases. Figures 7.1 and 7.2 illustrate the effect the homogeneity degree has in all strategies. We can see in the Kiviat graphs the evolution of a clearly suboptimal shape when we have a low homogeneity to a star-like shape when we have a high homogeneity. This indicates that the major difficulty for recommending for a group is its homogeneity, not its size.

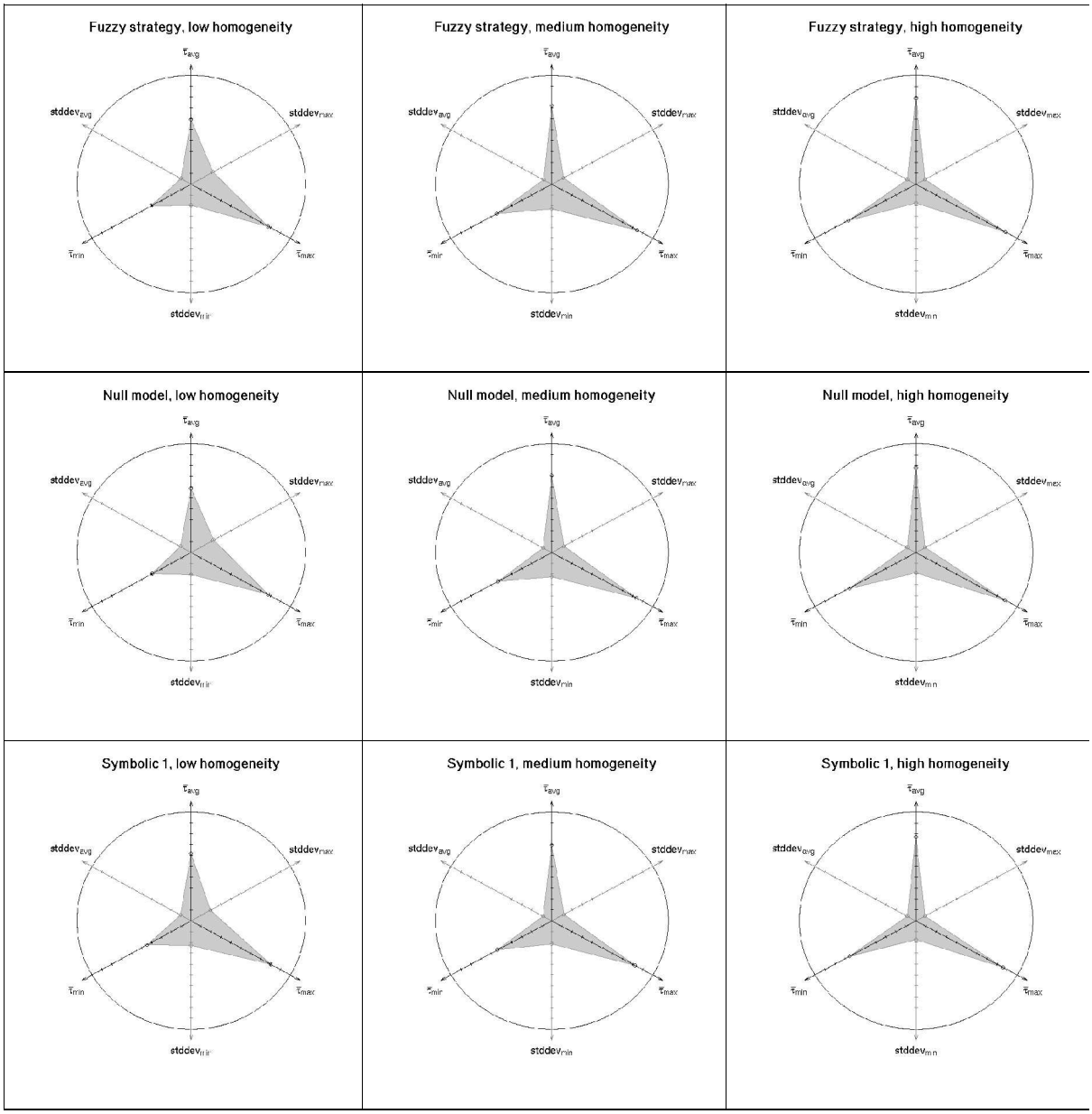


Figure 7.1 Effects of the homogeneity degree on the *Fuzzy*, *Null* and *Symbolic 1* strategies. Notice that as we progress from low to high homogeneity, the methodologies get nearer to the ideal "star shaped" area.

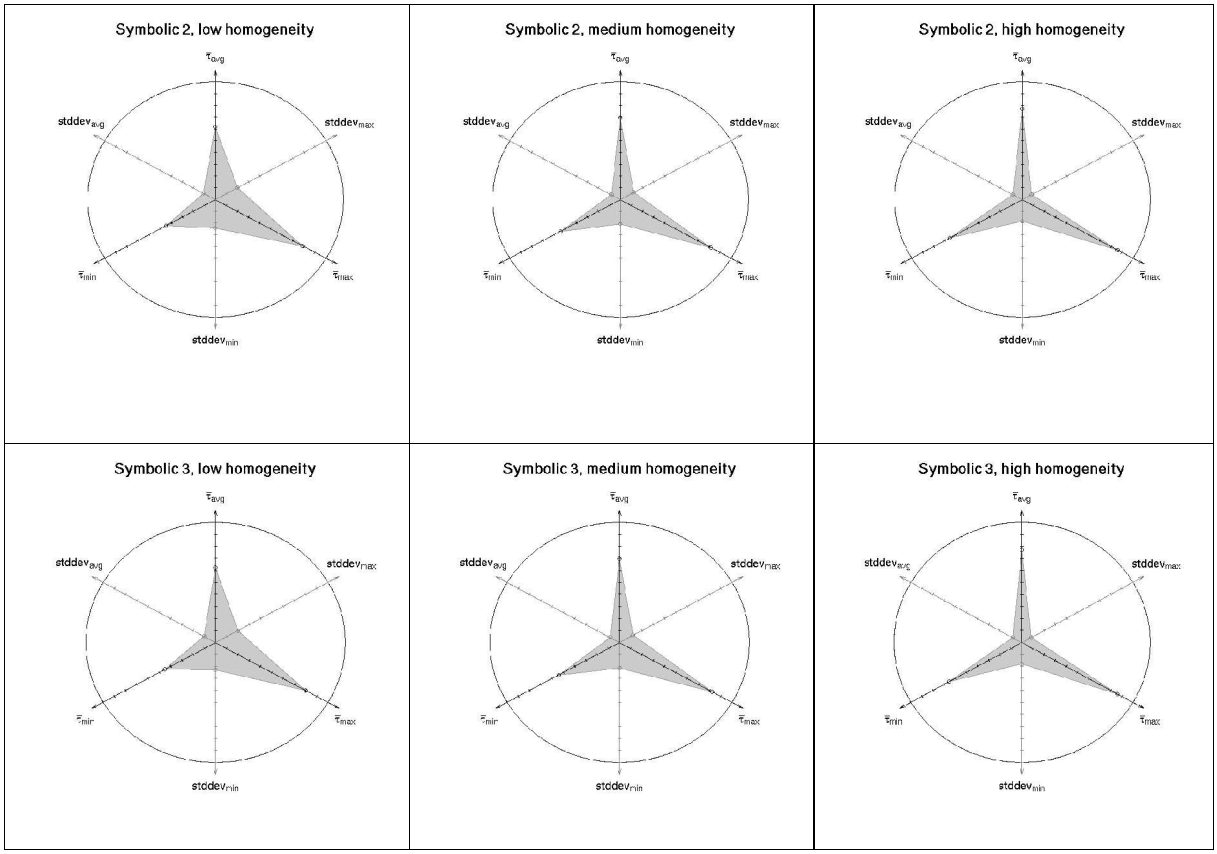


Figure 7.2 Effects of the homogeneity degree on the S*ymbolic 2* and *Symbolic 3* strategies. Notice that as we progress from low to high homogeneity, the methodologies get nearer to the ideal "star shaped" area.

Chapter 8

Conclusions

Contributions of this work, difficulties and some possible future developments.

8.1 Conclusions

The field of recommender systems for groups is young and offers many challenges to be explored. As we could see in this work, it is a multidisciplinary area, that involves several sciences like mathematics, psychology, social choice, operational research (and multicriteria decision making), as well as computer science. New technologies that will be (most likely) adopted in the near future, like the interactive TV, will generate a strong demand for personalization technologies, including recommendations for groups.

Being a problem both almost fresh in the field of recommender systems but very intermingled with problems well treated by the most diverse research areas, it is a difficult problem to approach: at the same time that we have almost no work available in the field of recommender systems; we have a vast amount of resource available in related fields. These fields often treat the problem from diverse points of view, requiring different background knowledge.

Despite the difficulties, we believe this work contributes to the field of recommender systems for groups. We can remark the following contributions:

- Identification of related research areas that treats similar problems. The work available in these areas (for example social psychology) should definitely be taken into account when trying to develop a recommender system for groups.
- Recognition of the potential of using fuzzy majority techniques for generating easier to explain recommendations. Although the simple different strategies we used did not behaved differently, this should be further explored. For example, by testing the ideas of Section 4.4.1.
- The possibility to use symbolic data analysis techniques in a new problem, the generation of recommendations (in special for groups), implementing principles of collaborative filtering using symbolic data analysis techniques to develop a novel model-based recommendation strategy for groups.
- The proposal of an evaluation framework to pragmatically measure the quality of group recommendations (including the importance of comparing the results with a baseline, "null" model).

8.2 Future work

Various new endeavors could be envisaged both in experimental as in theoretical views:

- The running of a live recommender system for groups with a large community of users would enable a new set of experimentations. For example, we could use learning to identify which recommendation strategies are most suitable for each group of users.
- A method of explaining the recommendations for groups may be developed. This method could even use artificial intelligence techniques to generate explanations in natural language.

- The automatic identification of "factions" inside highly heterogeneous groups, suggesting their division.
- We have not attempted to do a rigorous treatment to the problem of group decision making. The decision sciences may bring important contributions to a more profound analysis of the problem. A vast amount of work is available in the field, that can be used to do theoretical characterizations of recommendations for groups, including the features presented by different recommendation methods. A formal work could also be done to establish the relation of recommendation for groups to the problem of group (or multicriteria) decision making. Are they the same problem? Or they have different characteristics?

Bibliography

[1] Aggarwal, Charu; Wolf, Joel; Wu, Kun-Lung; Yu, Philip. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, USA, pages 201-212, August 1999.

[2] Agrawal, Rakesh; Imielinski, Tomasz; Swami, Arun. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., USA, pages 207-216, May 1993.

[3] Arrow, Kenneth J. *Social Choice and Individual Values*. Wiley, New York, USA, 2nd edition, 1963.

[4] Balabanovic, Marko; Shoham, Yoav. Fab: content-based, collaborative recomendation. *Communications of the ACM*. 40(3):66-72, March 1997.

[5] Barba-Romero, Sergio; Pomerol, Jean-Charles. *Decisiones Multicriterio: Fundamentos Teóricos y Utilización Práctica*. Universidad de Alcalá, Alcalá, Spain, 1st edition, 1997.

[6] Billsus, Daniel; Pazzani, Michael J. Learning collaborative information filters. In *Proceedings of the 15th International Conference on Machine Learning*, San Francisco, CA, USA, pages 46-54, 1998.

[7] Bock, Hans-Hermann; Diday, Edwin (eds.). *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer, Berlin-Heidelberg, Germany, 1st edition, 2000.

[8] Bondy, John A.; Murty, U. S. R. *Graph theory with applications*. Macmillan, North Holland, USA, 2nd edition, 1979.

[9] Breese, Jack S.; Heckerman, David; Kadie, Carl. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, Madison, WI, USA, pages 43-52, July 1998.

[10] Burke, Robin. The wasabi personal shopper: A case-based recommender system. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-99) and of the 11th Conference on Innovative Applications of Artificial Intelligence (IAAI-99)*, Menlo Park, CA, USA, pages 844-849, July 1999.

[11] Carenini, Giuseppe; Smith, Jocelyin; Poole, David. Towards more conversational and collaborative recommender systems. In *Proceedings of the 2003 International Conference on Intelligent User Interfaces (IUI-03)*, Miami, Florida, USA, pages 12-18, 2003.

[12] Chiclana, F.; Herrera, F.; Herrera-Viedma, E.; Poyatos, M. C. A classification method of alternatives for multiple preference ordering criteria based on fuzzy majority. *Journal of Fuzzy Mathematics*. 4(4):801-813, December 1996.

[13] Compaq Systems Research Center *Eachmovie collaborative filtering data set*. http://www.research.compaq.com/SRC/eachmovie/ , 2001.

[14] Cotter, Paul; Smyth, Barry PTV: Intelligent Personalised TV Guides. In *Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00) and of the 12th Conference on Innovative Applications of Artificial Intelligence (IAAI-00)*, Menlo Park, CA, USA, pages 957-964, July 2000.

[15] Diday, Edwin. Concept and Galois Lattices in Symbolic Data Analysis (talk). In *Proceedings of the Fourth International Conference Journées de l'Informatique Messine (JIM'2003)*, Metz, France, pages 71-80, September 2003.

[16] Dwork, Cynthia; Kumar, Ravi; Moni, Naor; Sivakumar, D. Rank Aggregation Methods for the Web. In *Proceedings of the Tenth International World Wide Web Conference (WWW10)*, Hong Kong, China, pages 613-622, May 2001.

[17] Fagin, Ronald; Kumar, Ravi; Sivakumar, D. Comparing top k lists. In *Proceedings of the 2003 ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, Baltimore, MD, USA, January 2003.

[18] Good, Nathaniel; Schafer, J. Ben; Konstan, Joseph A.; Borchers, Al; Sarwar, Badrul; Herlocker, Jon; Riedl, John. Combining Collaborative Filtering with Personal Agents for Better Recommendations. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-99)*, Menlo Park, CA, USA, pages 18-22, July 1999.

[19] Herlocker, Jonathan L. Understanding and Improving Automated Collaborative Filtering Systems. *Ph.D. Thesis*, Computer Science Dept., University of Minnesota, 2000.

[20] Herlocker, Jonathan L.; Konstan, Joseph A.; Borchers, Al; Riedl, John. An Algorithm Framework for Performing Collaborative Filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkley, CA, USA, pages 230-237, 1999.

[21] Herlocker, Jonathan L.; Konstan, Joseph A.; Riedl, John. Explaining Collaborative Filtering Recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work (CSCW '00)*, Philadelphia, PA, USA, pages 241-250, December 2000.

[22] Hill, Will; Stead, Larry; Rosenstein, Mark; Furnas, George. Recommending and evaluating choices in a virtual community of use. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, Denver, Colorado, USA, pages 194-201, May 1995.

[23] Hillier, Frederick S.; Lieberman, Gerald J. *Introduction to operations research*. Holden-Day, San Francisco, CA, USA, 1st edition, 1967.

[24] Hinsz, Verlin B. Group decision making with responses of a quantitative nature: The theory of social schemes for quantities. *Organizational Behavior and Human Decision Processes*. 80(1):28-49, October 1999.

[25] Jain, A. K.; Murty, M. N.; Flynn, P. J. Data Clustering: A Review. *ACM Computing Surveys*. 31(3):264-323, September 1999.

[26] Jain, Raj. *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, New York, USA, 1st edition, 1991.

[27] Kameda, Tatsuya; Tindale, R. Scott; Davis, James. Cognitions, preferences, and social sharedness: past, present, and future directions in group decision making. In Sandra L. Schneider and James Shanteau, editor, *Emerging Perspectives on Judgment and Decision Research*, chapter 14. Cambridge University, Cambridge, UK, 2003.

[28] Karypis, George Evaluation of Item-Based Top-N Recommendation Algorithms. In *Proceedings of the Tenth International ACM Conference on Information and Knowledge Management (CIKM-01)*, Atlanta, Georgia, USA, pages 247-254, November 2001.

[29] Kaufman, Leonard; Rousseeuw, Peter J. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, New York, USA, 9th edition, 1990.

[30] Kendall, Maurice. *Rank Correlation Methods*. Charles Griffin & Company, High Wycombe, Bucks, UK, 4th edition, second impression, 1975.

[31] Kitts, Brendan; Freed, David; Vrieze, Martin. Cross-sell: a fast promotion-tunable customer-item recommendation method based on conditionally independent probabilities. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-00)*, Boston, MA, USA, pages 437-446, August 2000.

[32] Krulwich, Bruce; Burkey, Chad. Learning user information interests through the extraction of semantically significant phrases. In *Proceedings of AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, CA, USA, March 1996.

[33] Levine, John M. Transforming individuals into groups: Some hallmarks of the SDS approach to small group research. *Organizational Behavior and Human Decision Processes*. 80(1):21-27, October 1999.

[34] Lieberman, Henry; Van Dyke, Neil W.; Vivacqua, Adrian S. Let's browse: A collaborative web browsing agent. In *Proceedings of the 1999 International Conference on Intelligent User Interfaces, Collaborative Filtering and Collaborative Interfaces (IUI-99)*, Los Angeles, CA, USA, pages 65-68, January 1999.

[35] Lin, Weiyang; Alvarez, Sergio A.; Ruiz, Carolina. Collaborative recommendation via adaptive association rule mining. In *Proceedings of the Web Mining for E-Commerce Workshop (WebKDD'2000)*, Boston, MA, USA, August 2000.

[36] Maes, Pattie. Agents that reduce work and information overload.

Communications of the ACM. 37(7):30-40, July 1994.

[37] Malone, Thomas W.; Grant, Kenneth R.; Turbak, Franklyn A.; Brobst, Stephen A.; Cohen, Michael D. Intelligent information-sharing systems. *Communications of the ACM*. 30(5):390-402, May 1987.

[38] Manber, Udi. *Introduction to Algorithms: A Creative Approach*. Addison-Wesley, Reading, MA, USA, 2nd printing, 1989.

[39] Meyer, Joachim; Gilat, Sharon; Erev, Ido. Consensus effects in categorization decisions. *Journal of Mathematical Psychology*. 47(4):417-428, August 2003.

[40] Miyahara, Koji; Pazzani, Michael J. Collaborative Filtering with the Simple Bayesian Classifier. In *Pacific Rim International Conference on Artificial Intelligence*, Melbourne, Australia, pages 679-689, 2000.

[41] Mohammed, Susan; Ringseis, Erika. Cognitive diversity and consensus in group decision making: the role of inputs, processes, and outcomes. *Organizational Behavior and Human Decision Processes*. 85(2):310-335, July 2001.

[42] Montgomery, Douglas C. *Design and Analysis of Experiments*. John Wiley & Sons, New York, USA, 4th edition, 1997.

[43] Oard, Douglas W.; Kim, Jinmook. Implicit feedback for recommender systems. In *Proceedings of the AAAI Workshop on Recommender Systems*, Madison, WI, USA, pages 81-83, July 1998.

[44] O'Connor, Mark; Cosley, Dan; Konstan, Joseph A.; Riedl, John. PolyLens: A recommender system for groups of users. In *Proceedings of the Seventh European Conference on Computer Supported Cooperative Work (ECSCW 2001)*, Bonn, Germany, pages 199-218, September 2001.

[45] Paulson, Patrick; Tzanavari, Aimilia. Combining collaborative and content-based filtering using conceptual graphs. In J. Lawry, J. Shanahan, and A. Ralescu, editor, *Lecture Notes in Artificial Intelligence. Modeling with Words*. Springer-Verlag, Berlin, Germany, to appear.

[46] Queiroz, Sérgio R. M.; De Carvalho, Francisco A. T.; Ramalho, Geber L.; Corruble, Vincent. Making Recommendations for Groups Using Collaborative Filtering and Fuzzy Majority. In *Proceedings of the 16th Brazillian Symposium on Artificial Intelligence (SBIA 2002), Lecture Notes in Artificial Intelligence (LNAI/LNCS) 2507*, Springer, Berlin-Heidelberg, Germany, pages 248-258, November 2002.

[47] Rashid, Al Mamunur; Albert, Istvan; Cosley, Dan; Lam, Shyong K.; McNee, Sean M.; Konstan, Joseph A.; Riedl, John. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the 2002 International Conference on Intelligent User Interfaces (IUI-02)*, San Francisco, CA, USA, pages 127-134, 2002.

[48] Resnick, Paul; Iacovou, Neophytos; Suchak, Mitesh; Bergstrom, Peter; Riedl, John. Grouplens: An open architecture for collaborative filtering of netnews.

In *Proceedings of the Ninth ACM Conference on Computer-Supported Cooperative Work*, Chapel Hill, North Carolina, United States, pages 175-186, October 1994.

[49] Ripley, Brian D. The R project in statistical computing. *MSOR Connections. The newsletter of the LTSN Maths, Stats & OR Network*. 1(1):23-25, February 2001.

[50] Ross, Kenneth A.; Wright, Charles R. B. *Discrete Mathematics*. Prentice Hall, Englewood Cliffs, News Jersey, USA, 2nd edition, 1988.

[51] Saari, Donald G.; Valognes, Fabrice. Geometry: Voting, and paradoxes. *Mathematics Magazine*. 78(October):243-259, 1998.

[52] Sarwar, Badrul, M.; Konstan, Joseph A.; Borchers, Al; Herlocker, Jon; Miller, Brad; Riedl, John. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the ACM 1998 conference on computer supported cooperative work*, Seattle, Washington, USA, pages 345-354, November 1998.

[53] Sarwar, Badrul; Karypis, George; Konstan, Joseph; Riedl, John. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Proceedings of the Fifth International Conference on Computer and Information Technology (ICCIT 2002)*, Dhaka, Bangladesh, December 2002.

[54] Sarwar, Badrul; Karypis, George; Konstan, Joseph; Riedl, John. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference (WWW10)*, Hong Kong, pages 285-295, May 2001.

[55] Sarwar, Badrul; Karypis, George; Konstan, Joseph; Riedl, John. Analysis of Recommendation Algorithms for E-Commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, Minneapolis, Minnesota, USA, pages 158-167, October 2000.

[56] Schafer, J. Ben; Konstan, Joseph; Riedl, John. Recommender Systems in E-Commerce. In *Proceedings of the ACM Conference on Electronic Commerce (EC-99)*, Denver, Colorado, USA, pages 158-166, November 1999.

[57] Shardanand, Upendra; Maes, Pattie Social information filtering: algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, Denver, Colorado, USA, pages 210-217, May 1995.

[58] Smyth, Barry; Cotter, Paul. A personalized television listings service. *Communications of the ACM*. 43(8):107-111, August 2000.

[59] Soboroff, Ian M.; Nicholas, Charles K. Combining Content and Collaboration in Text Filtering. In *Proceedings of the IJCAI'99 Workshop on Machine Learning in Information Filtering*, Stockholm, Sweden, pages 86-91, August 1999.

[60] Stasser, Garold. A primer of social decision scheme theory: Models of group

influence, competitive model-testing, and prospective modeling. *Organizational Behavior and Human Decision Processes*. 80(1):3-20, October 1999.

[61] Stroud, Jim. TV personalization: A key component of interactive tv. *Technical report.* The Carmel Group, 2001. Available at http://www.carmelgroup.com.

[62] Struyf, Anja; Hubert, Mia; Rousseeuw, Peter J. Clustering in an object-oriented environment. *Journal of Statistical Software*, 1, 1996. http://www.jstatsoft.org

[63] Terveen, Loren; Hill, Will. Human-computer collaboration in recommended systems. In John M. Carroll, editor, *Human-Computer Interaction in the New Millennium*, chapter 22. Addison Wesley, Boston, MA, USA, 2002.

[64] Terveen, Loren; Hill, Will; Amento, Brian; McDonald, David; Creter, Josh. PHOAKS: A system for sharing recommendations. *Communications of the ACM*. 40(3):59-62, March 1997.

[65] Yager, Ronald R. Families of OWA operators. *Fuzzy Sets and Systems*. 59(2):125-148, October 1993.