Dissertation

Recommendation and Decision Support Techniques for Groups

MSc. Thi Ngoc Trang Tran, BSc

Graz, November 2019

Institute for Software Technology
Graz University of Technology

Supervisor/First reviewer: Univ.-Prof. Dipl.-Ing. Dr.techn. Alexander Felfernig
Second reviewer: Assoc.Prof. Mag. Dr. Gerhard Leitner

Abstract (English)

In recent years, recommending products or services to groups has been the subject of many studies in recommender systems since many decisions in daily life are more likely made by groups of users rather than single users, for example, selecting a movie to watch with friends, choosing a restaurant to have dinner with colleagues, or deciding on a destination to visit with family members.

Group recommender systems are genuinely beneficial for users since they facilitate a group decision making process and provide group recommendations that help to satisfy group members. Group recommendations are usually generated based on aggregation strategies. These strategies merge the preferences of all individual group members into a group profile, which represents the inferred preference of the whole group. However, these approaches do not always provide group recommendations that satisfy the preferences of all group members. Also, social aspects such as *fairness* or *consensus* among group members are not considered. On the other hand, techniques to support groups of users to jointly make decisions on complex products (i.e., products which are characterized by a set of attributes) and solutions to resolve conflicts between group members' preferences are missing. Moreover, group recommender systems have been suffering from decision biases that can deteriorate the quality of decision outcomes.

In this thesis, we propose different decision support techniques to resolve the aforementioned issues and target at improving the efficiency of group decision making processes as well as the quality of decision outcomes. One of the proposed techniques is *group-based configuration*, which supports groups of users to jointly configure complex products/services and to resolve inconsistencies between group members' preferences or between group members' preferences and the knowledge base. Also, in the context of group-based configuration, we propose a solution based on the concept of *liquid democracy* to deal with the insufficient knowledge of group members and therefore to improve the quality of preference acquisition.

Besides, to increase the trust and acceptance of users concerning group recommendations, we propose different approaches to generate explanations of recommended items. The proposed explanations provide insights into the underlying mechanisms of why a specific item has been recommended to the group. Moreover, these explanations consider the fairness perception, consensus perception, and satisfaction of users with regard to group recommendations.

Finally, we investigate decision biases in group decision making processes (such as *serial position effects* and *decision manipulation*) and propose solutions to counteract these. *Serial position effects* usually trigger decision biases (when items are represented in the form of a list). In this thesis, we investigate serial position effects occurring when a *given sequence of decisions* is made by group members. These decisions can be related to *low-* as well as *high-involvement item domains* with low and high related decision efforts.

Besides, they are continuously made by group members within a certain amount of time. In this context, we examine if group members use different decision making strategies for the same decision depending on its position in the sequence. *Decision manipulation* in group recommender systems can be referred to as *an attack* in which some group members try to strategically adapt their preferences for items to push their favorite options. This action can result in serious vulnerabilities to the quality of group recommendations and decrease the trust of users in group recommender systems. To counteract this bias, we propose user interfaces which represent the rating adaptation history of all group members. We show that this approach helps to effectively mitigate the decision manipulation issue in group recommender systems.

Abstract (German)

Empfehlungssysteme mit dem Fokus der Bereitstellung von *Produkten oder Dienstleistungen* für eine Gruppe von Personen, wurden in den letzten Jahren immer populärer. Dies liegt vor allem daran, dass viele Entscheidungen im Alltag nicht von Einzelpersonen, sondern von einer Gruppe von Menschen getroffen werden. Beispiele dafür sind die Wahl eines Restaurants für ein Abendessen mit Kollegen, die Wahl eines Filmes für einen Filmabend mit Freunden oder die Bestimmung eines Reiseziels für einen Familienurlaub.

Im Allgemeinen ermöglichen Gruppenempfehlungstechnologien, Entscheidungen zu identifizieren, welche die Zufriedenheit aller Gruppenmitglieder maximieren. Hierfür werden Aggregationsstrategien verwendet, um die Präferenzen der einzelnen Gruppenmitglieder zusammenzuführen und daraus ein Gruppenprofil abzuleiten. Solche Verfahren sind aber nicht immer in der Lage, die Zufriedenheit einzelner Gruppenmitglieder zu garantieren, da oft soziale Aspekte, wie zum Beispiel *Fairness* oder *Konsens unter den Gruppenmitgliedern*, vernachlässigt werden. Andererseits werden Techniken zur Unterstützung von Benutzergruppen bei der gemeinsamen Entscheidungsfindung für komplexe Produkte sowie Möglichkeiten zur Lösung von Konflikten zwischen den Präferenzen der Gruppenmitglieder nicht ausreichend berücksichtigt. Darüber hinaus leiden viele Gruppenempfehlungstechnologien oft auch unter einem Bias, welcher die Qualität der getroffenen Entscheidungen negativ beeinflussen kann.

In unserer Arbeit stellen wir verschiedene Ansätze für Gruppenempfehlungssysteme vor, die darauf ausgelegt sind, die aufgezeigten Probleme zu lösen, sowie die Effizienz des Entscheidungsfindungsprozesses und die Qualität der getroffenen Entscheidungen zu verbessern. Die von uns entwickelten *Konfigurationstechnologien* ermöglichen es beispielsweise einer Gruppe von Personen, kollaborativ eine Konfiguration eines Produkts oder einer Dienstleistung zu erstellen und Konflikte selbst zu lösen. Darüber hinaus stellen wir das Konzept der *Liquid Democracy* vor, welches mangelndes Domänenwissen einzelner Gruppenmitglieder ausgleicht und dadurch die Qualität der Präferenzfindung verbessert.

Um das Vertrauen und die Akzeptanz von Gruppenempfehlungstechnologien zu steigern, stellen wir verschiedene Ansätze vor, welche die Vorschläge des Systems auch erklären. Diese Erklärungen zeigen, warum bestimmte Vorschläge getätigt werden und schließen dabei verschiedene soziale Aspekte der Gruppenmitglieder mit ein (z.B. Gerechtigkeit oder Konsens-Verständnis). Dadurch kann die Benutzerzufriedenheit mit Gruppenempfehlungstechnologien gesteigert werden.

Abschließend befassen wir uns mit sog. *"Decision Biases"* im Zusammenhang mit der Entscheidungsfindung in Gruppen. Dazu zählen unter anderen *der serielle Positionseffekte* und *Entscheidungsmanipulationen*, denen wir mit unserer Arbeit entgegenwirken. *Serielle Positionseffekte* treten häufig auf, wenn Auswahlmöglichkeiten in Listenform vorliegen. Wir beschäftigen uns konkret mit seriellen Positionseffekte, die bei der Entscheidungsfindung in *einer Kette von Einzelentscheidungen* entstehen. Dabei können Einzelentscheidungen verschiedenen Domänen, die durch unterschiedliche Komplexitätsgrade geprägt sind, angehören und müssen von Gruppenmitgliedern innerhalb von bestimmten zeitlichen Rahmenbedingungen getroffen werden. Wir untersuchen dabei, ob Gruppenmitglieder verschiedene Entscheidungsstrategien verfolgen, abhängig davon, wann einzelne Entscheidungen getroffen werden müssen. *Entscheidungsmanipulationen* treten dann auf, wenn einzelne Gruppenmitglieder versuchen, ihre Präferenzen derart zu adaptieren, dass das von ihnen favorisierte Resultat eintritt. Dies hat oft zur Folge, dass die Qualität und das Vertrauen in Empfehlungstechnologien gemindert werden. Um Manipulationen entgegenzuwirken, schlagen wir Benutzeroberflächen vor, die das Entscheidungsverhalten von Benutzern transparent darstellen.

Acknowledgement

I have been so lucky to have a chance to pursue my Ph.D. at the Institute of Software Technology, Graz University of Technology, Graz, Austria. During my four years studying here, I have learned a lot of amazing experiences that really change my outlook, and I would not make it possible without the support and guidance of many people around me.

Firstly, I would like to express my gratitude to my supervisor, Professor Alexander Felfernig, for his continuous support, his patience, his motivation, and his immense knowledge. He has motivated me to challenge my knowledge, inspired me, and brought me to interesting fields of research. His guidance actually helped me unveil my insight, develop my thinking, and focus on my studies to write this thesis.

Secondly, I would also like to thank the second reviewer of my dissertation, Professor Gerhard Leitner, for his insightful comments that help me to improve my thesis.

Thirdly, I would like to send my gratitude to my colleagues from the Applied Artificial Intelligence research group, Müslüm Atas, Seda Polat Erdeniz, Martin Stettinger, and Ralph Samer, and my colleagues from Institute of Software Technology who have made many contributions to my studies. They not only cooperated closely with me but also gave in their insightful discussions to help me develop my studies more detailed and clearer.

Also, I would like to thank the Vietnamese Government and the OeAD organization for their financial support and their assistance during my studies in Austria. Especially, I would like to express my gratefulness to Ms. Christina Düss in the Regional Office Graz who has been helping me to resolve financial and health-insurance issues during the past four years.

Additionally, I am so grateful to my uncle Chung Le, my aunts Helen Nguyen and Vui Tran, and my friends Hoa Le, Matthias Wölbitsch, and Julia Katharina Binder who have proofread my dissertation at different stages of completion. Also, my Vietnamese and Austrian friends in Graz have always been close to me and encouraged me whenever I face difficulties in life and research. Without their support, it must have been hard for me to overcome this rough time.

Lastly, I am deeply grateful to my parents and my siblings for their spiritual support throughout the past four years. Most profoundly, I owe a huge debt of gratitude to my parents-in-law and my husband. They have taken care of my son and everything back home for me so that I could focus well on my studies. Without my family's support, nothing would have been possible for me. Especially, I reserve my very last thanks to my son, my Sam. He is always the one I have been thinking of first, and he is always the one who has been giving me the best motivations to make me overcome difficulties during my studies here in Graz.

Thi Ngoc Trang Tran

Graz, 2019

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Graz, ______________________ ______________________

Place, Date Signature

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Text-dokument ist mit der vorliegenden Dissertation identisch.

Graz, am ______________________ ______________________

Ort, Datum Unterschrift

Contents

1. Introduction **1**
1.1. Background and Motivation . . . 1
1.2. Research Questions . . . 4
1.3. Thesis Contributions . . . 9
1.4. Thesis Outline . . . 12

2. Towards Group-based Configuration **15**
2.1. Abstract . . . 15
2.2. Introduction . . . 15
2.3. Group-based Configuration . . . 16
2.4. Resolving Inconsistencies in Group Preferences . . . 17
2.5. Future Work . . . 20
2.6. Conclusion . . . 22

3. Liquid Democracy in Group-based Configuration **23**
3.1. Abstract . . . 23
3.2. Introduction . . . 23
3.3. Working Example . . . 25
3.4. Application of Liquid Democracy . . . 26
3.5. Application of Multi-attribute Utility Theory . . . 28
3.6. Release Planning . . . 30
3.7. Conclusion and Future Work . . . 31

4. Explanations for Groups **33**
4.1. Abstract . . . 33
4.2. Introduction . . . 33
4.3. Explanations in Recommender Systems for Single Users and Groups . . . 34
4.3.1. Explanations in Single-User Recommender Systems . . . 34
4.3.2. Explanations in Group Recommender Systems . . . 35
4.4. Collaborative Filtering . . . 36
4.5. Content-based Filtering . . . 38
4.6. Constraint-based Recommendation . . . 41
4.7. Critiquing-based Recommendation . . . 47
4.8. Conclusion and Research Issues . . . 48

5. Towards Social Choice-based Explanations in Group Recommender Systems 49
5.1. Abstract 49
5.2. Introduction 49
5.3. Related work 51
5.4. Social Choice-based Explanations 52
5.4.1. Social Choice-based Preference Aggregation Strategies 52
5.4.2. Social Choice-based Explanations 53
5.5. Hypotheses and User Study 56
5.5.1. Hypotheses 56
5.5.2. User Study Design 56
5.6. Data Analysis Results and Discussions 58
5.6.1. Hypothesis H_1 58
5.6.2. Hypothesis H_2 61
5.6.3. Hypothesis H_3 62
5.7. Conclusion and Future work 64

6. User Interfaces for Counteracting Decision Manipulation In Group Recommender Systems 67
6.1. Abstract 67
6.2. Introduction 67
6.3. Related Work 68
6.4. User Interfaces for Counteracting Decision Manipulation 69
6.4.1. Rating Adaptation Information 69
6.4.2. User Interfaces for Counteracting Decision Manipulation 70
6.5. Research Questions and User Study 75
6.5.1. Research Questions 75
6.5.2. User Study 78
6.6. Data Analysis and Results 79
6.6.1. Data Analysis 79
6.6.2. Results 79
6.7. Conclusion and Future Work 84

7. Recommender Systems in the Healthy Food Domain 85
7.1. Abstract 85
7.2. Introduction 85
7.3. Recommender Systems 86
7.3.1. Recommendation Techniques for Individuals 86
7.3.2. Recommendation Techniques for Groups 89
7.4. Food Recommender Systems 90
7.4.1. Type 1: Considering User Preferences 91
7.4.2. Type 2: Considering Nutritional Needs of Users 94
7.4.3. Type 3: Balancing between User Preferences and Nutritional Needs of Users 98
7.4.4. Type 4: Food Recommender Systems for Groups 99
7.5. Research challenges 102
7.5.1. Challenges regarding User Information 103
7.5.2. Challenges regarding Recommendation Algorithms 104
7.5.3. Challenges regarding Changing Eating Behavior of Users 105
7.5.4. Challenges regarding Explanations 105
7.5.5. Challenges regarding Group Decision Making 105
7.6. Conclusion 106

8. Investigating Serial Position Effects in Sequential Group Decision Making **107**
8.1. Abstract 107
8.2. Introduction 107
8.3. Group Aggregation Strategies 108
8.4. Hypotheses and User Study 109
8.4.1. Hypotheses 109
8.4.2. User study 110
8.5. Data Analysis Results and Discussions 112
8.6. Conclusions and Future Work 114

9. Conclusions & Future Work **115**
9.1. Conclusions 115
9.2. Future Work 119
9.2.1. The Influence of Group Dynamics on Group Decision Making 119
9.2.2. Explanations and Visualizations for Supporting Consensus 121
9.2.3. Persuasive Explanations 122
9.2.4. Further Group Recommendation Approaches 123

List of Figures **125**

List of Tables **127**

Bibliography **131**

Chapter 1

Introduction

1.1. Background and Motivation

Nowadays, heavy information overload hinders Internet users in finding useful and necessary information. In this context, recommender systems have been recognized as effective tools to help users overcome such obstacles by selecting valuable information from a huge amount of available data sources (Kapoor, 2017). Recommender systems are usually employed in different domains to recommend products/services that meet users' needs and preferences (Burke, 2000; Burke et al., 2011; Tran et al., 2018a), such as *movies* (Gomez-Uribe and Hunt, 2015), *music* (Lu and Tintarev, 2018), *books* (Linden et al., 2003), *tourism destinations* (Borrís et al., 2014), and *financial services* (Felfernig, 2016). An example of book recommender systems is shown on the popular e-commerce pages of *Amazon.com*. In this system, recommendation algorithms are applied to personalize the online store for each customer based on his/her interests, e.g., showing the title of the books regarding programming languages to a software developer or presenting baby toys to a young mother (Linden et al., 2003). In recent years, researchers have paid their attention to the *healthcare* domain to resolve issues that users have to face when looking for helpful information from a huge set of *health-related data sources* (Wiesner and Pfeifer, 2014). Health recommender systems help patients better understand their health situations, recommend medical remedies, and offer solutions that motivate them to follow healthy lifestyles (Wiesner and Pfeifer, 2014; Schäfer et al., 2017). One example thereof is a food recommender system which helps to deal with *malnutrition issues* of the elderly (Aberg, 2006). This system provides users with a *menu-planning tool* which facilitates the menu generation process for old people. The created menus take into account all *user-related information*, such as food preferences, dietary restrictions, nutritional values, preparation time, the availability of ingredients, and the variety of meals in terms of used ingredients and meal categories (Aberg, 2006).

Recommendations are usually generated based on the following approaches, which are denoted as *collaborative filtering, content-based, knowledge-based*, and *hybrid*. *Collaborative filtering* makes recommendations to an active user based on the most preferred items of his/her *nearest neighbors* (i.e., users whose preferences are most similar to the active user) (Goldberg et al., 1992). *Content-based* approaches recommend items that are similar to the ones consumed by the user in the past (Ricci et al., 2010; Lops et al., 2011). *Knowledge-based* approaches are usually applied to generate recommendations in domains where the quantity of available item ratings is quite limited (such as *cars*, *apartments*, and *financial services*) or when the user wants to explicitly define his/her requirements for items (e.g., *"the apartment should be close to working area"*). These approaches generate recommendations based on *the knowledge about the items*, *explicit user preferences*, and *a set of constraints* describing the dependencies between users' preferences and items' properties (Felfernig and Burke, 2008). Finally, *hybrid* approaches are a combination of the aforementioned approaches for the sake of using the advantages of one approach and fixing the disadvantages of another approach (Ricci et al., 2010). For instance, collaborative filtering approaches usually face the *new-item* issue triggered when a new item is added to the system and no user has rated it, whereas content-based approaches can tackle this problem since the prediction for new items is generally based on available descriptions of these items.

While previously published studies on recommender systems are limited to single-user decisions (Masthoff, 2011; Kapcak et al., 2018), there are plenty of scenarios in reality where decisions are more likely made by groups of users instead - for instance, choosing a list of songs to be played in a fitness studio (McCarthy and Anagnost, 1998) or deciding on a tourism destination for summer holiday (Ardissono et al., 2003). In this context, *group recommender systems* have emerged as powerful tools to support group decision making processes and generate group recommendations based on models of all group members. These models are usually constructed by aggregating the preferences of all individual group members. Existing research has shown two main approaches to aggregate the preferences of individual group members: *aggregated predictions* and *aggregated models* (Jameson and Smyth, 2007; Felfernig et al., 2018a). *Aggregated predictions* first generate recommendations for individual group members and then propose group recommendations based on merging these recommendations. *Aggregated models* combine all individual group members' preferences into a *group preference model (i.e., group profile)* which represents the inferred preferences of the whole group. The group profile is then applied to generate group recommendations (Felfernig et al., 2018a). The *aggregated models* approach is helpful in scenarios in which group members want to perform additional actions in the group decision making process, such as *analyzing, negotiating,* or *adapting the preferences of the group* (Jameson and Smyth, 2007). Furthermore, this approach helps to conserve the privacy of group members since individual group members' profiles are not recorded in the system (Felfernig et al., 2018a).

Compared to individual recommendations, making group recommendations is more complex. Even though we know very well what is suitable for each group member, how to combine individual user models is quite challenging (Masthoff, 2011). The reason is that the combination of individual user models should be done in such a way that the preferences of group members are taken into account as far as possible. In other words, this combination is not merely the sum of individual group members' preferences. Rather than that, it should be the aggregation of all individual group members' preferences on the basis of considering the *social aspects* within the group, such as *"how to help the group agree on a decision"* (Chiclana et al., 2007), *"how to resolve conflicts among group members"* (Felfernig et al., 2016), and *"how to foster fairness among group members"* (Kacprzyk and Zadrożny, 2016). Taking into account specific social aspects can help to increase the satisfaction of users with regard to group recommendations.

Within a group decision making process, group members might face situations in which they have to jointly make a decision on *complex products* characterized by many features. For instance, when deciding on a *tourism package* for the upcoming summer holiday, group members have to jointly specify different features, such as *"where to go"* (i.e., destinations), *"what to do"* (i.e., activities at the destination), *"where to stay"* (i.e., accommodation), *"what to eat"* (i.e., food), *"how to get around"* (i.e., means of transportation), and *"how much to pay"* (i.e., cost/price) (Tran et al., 2016). In this scenario, difficulties might arise when some group members do not have enough experience/knowledge to precisely specify/evaluate all the necessary features, or they might be unsure about their preferences for some features of the tourism package. *Configuration systems* can be beneficial tools to tackle such an issue. Similar to knowledge-based recommender systems, configuration systems support users to specify their requirements for products/services and provide useful feedback. Feedback in configuration systems can be *further questions* that need to be answered, *solutions* (configurations), *explanations of solutions*, and *adaptation* *solutions* when no solution can be found (Falkner et al., 2011).

Configuration systems also need recommendation techniques to ease the configuration process. For instance, Pereira et al. (2018) proposed *feature-based recommendation techniques* that guide the user to valid and relevant parts of the configuration process. Falkner et al. (2011) applied recommendation technologies to exclude features that are not necessary to the user in a specific context and also to rank the features so that the user easily accesses the most relevant ones. Besides, in scenarios where the user's requirements are *inconsistent* with product knowledge, these authors also proposed *minimal explanations* showing minimal sets of requirements to be adapted or deleted by the user so that a solution could be identified (Reiter, 1987; Felfernig et al., 2009). In another research, Coster et al. (2002) proposed an approach to recommending feature values to users in order to help them have a better understanding of the dependencies between requirements and possible settings of other variables. Most of the configuration technologies are typically applied in *closed settings* where knowledge bases are developed by a single user or a small group of users (Felfernig et al., 2014b), whereas configuration techniques for *group settings* have received scant attention. Some open questions have arisen in this context, such as *"how to support configuration for groups?"* and *"how to resolve inconsistencies in group-based configuration scenarios?"*.

In the end of the group decision making process, *decision outcomes* (i.e., *recommended items*) are generated and then sent to users. However, the recommended items are usually shown to the users in the form of *"black boxes"* which prevent them from comprehending recommended results, such as *"why these items have been selected to users"* or *"how they have been generated"*. This raises difficulties for users to decide if they can trust the recommended items without inspecting all of them in detail (Gedikli et al., 2014). In many studies on recommender systems for single users, explanations have been denoted as additional information that helps users understand mechanisms behind the recommendation process (i.e., *transparency*), choose better solutions (i.e., *scrutability*), speed up the decision making process (i.e., *efficiency*), and increase their trust and acceptance with regard to recommended items (i.e., *trust* and *persuasiveness*) (Tintarev and Masthoff, 2007; Jannach et al., 2010). In the context of group recommendations, explanations can bring a significant impact on how group members perceive recommended items, for instance, increasing the perceived understandability and the satisfaction of group members with the final group decision (Stettinger et al., 2015). Furthermore, compared to the explanations for single users, the explanations for a group of users could have further goals. Some examples thereof are *detecting conflicts* among group members, *enhancing mutual awareness* within the group, *increasing the fairness or consensus perception* of users concerning recommended items, and *accelerating the consensus making process*.

When making a decision, instead of optimizing the decision, users tend to apply *cognitive heuristics* - *mental shortcuts* that allow users to simplify their complex thought processes. Thereby, they help users come to a decision faster with less mental investment (Kahneman, 2011). In the context of group decision making scenarios, group members are also assumed to use *cognitive heuristics* which cause different *decision biases* (Felfernig et al., 2018b). These biases can deteriorate the input quality of recommender systems and lead to suboptimal decision outcomes. Therefore, group recommender systems should be able to detect such biases and counteract their negative impacts on the quality of decision outcomes.

There exist different types of decision biases that can be recognized in group decision making processes (for details on these biases, we refer to (Felfernig et al., 2018b)). One of the most well-known decision biases is *serial position effects* referred to as *cognitive phenomena* occurring when items are presented in the form of a list (Mandl et al., 2011). When determining relevant items, users usually do not want to evaluate an extensive list of items. Instead, they tend to focus on the evaluations of items shown *at the beginning* and *at the end* of a list, whereas those shown in the middle of the list seem to be ignored

(Murphy et al., 2017). In the context of *multi-attribute items* in which each item is characterized by a set of attributes, Felfernig et al. (2007a) confirmed that item attributes presented at the beginning and the end of a recommendation dialog were more likely to be recalled compared to remaining attributes. Besides, these attributes were also considered as selection criteria when choosing items from a consideration set.

Decision manipulation has been detected as another source of biased decision making. Decision manipulation was first detected in e-commerce sites as *shilling attacks* where malicious users with carefully chosen profiles are *injected* into the system to push the predictions of some target items (Van Roy and Yan, 2009; Li and Luo, 2011; Zhou et al., 2015). In group recommender systems, decision manipulation can be referred to as rating adaptations of some group members for items to push their favorite options. This decision manipulation can be triggered when *user-control mechanisms* (McNee et al., 2003; Xiao and Benbasat, 2007; Jannach et al., 2017) are implemented in group recommender systems, such as *allowing users to articulate their preferences for items, to see others' preferences*, or *to change their preferences for items to achieve a consensus* (McCarthy et al., 2006; Palomares et al., 2014a; Stettinger et al., 2015). In this context, some users might take advantage of the user controls to insincerely report their preferences, as this will return an outcome that they preferred (Jameson et al., 2003; Conitzer and Yokoo, 2010). For instance, in a group decision making scenario among three users, *user 1* rated item *X* with *3 stars* (in a *5-star* rating scale). Thereafter, he/she saw two other group members (*user 2* and *user 3*) respectively rated this item with *5 stars* and *3 stars*. The average rating of *item X* is now equal to *4 stars* (after rounding up). In order to ensure that the average rating of this item will be *3 stars* as he/she intended, *user 1* decreased the ratings of this item from *3 stars* down to *1 star* (Jameson et al., 2003). The decision manipulation in this scenario can be referred to as *an attack* in which a group member tries to strategically adapt the ratings of some items to his/her own advantage. This can result in serious vulnerabilities to the quality of group recommendations. Therefore, in order to conserve the quality of decision outcomes, group recommender systems should be aware of such manipulations and implement mechanisms to counteract these (Jameson et al., 2003; Conitzer and Yokoo, 2010; Stettinger et al., 2015).

1.2. Research Questions

Based on the previous discussions, this thesis attempts to address the following research questions (see also Table 1.1):

1. **Support configuration for groups**

   Existing configuration techniques are usually applied to *basic settings* in which knowledge bases and corresponding configurations are developed by a single user or a small group of users (Felfernig et al., 2014b). These techniques cause issues concerning *scalability* and *suboptimal decisions*. *Scalability problems* indicate a *knowledge acquisition bottleneck* (Hayes-Roth et al., 1983; Felfernig et al., 2014b) occurred in knowledge engineering when the domain knowledge is transferred into a configuration knowledge base. To resolve such an issue, different approaches were proposed to improve the efficiency of knowledge engineering, such as *graphical knowledge engineering* (Felfernig et al., 2000a) and *intelligent debugging* (Felfernig et al., 2000b; Schubert et al., 2010; Felfernig et al., 2012a). However, these solutions face the problem of *missing scalability*, which especially becomes manifest when the complexity of configuration knowledge bases exceeds the available resources for performing corresponding developments and maintenance operations (Felfernig et al., 2014b). *Suboptimal decisions* are triggered in scenarios where a single user has to configure the requirements and preferences of the whole group. These configurations might not reflect the group preferences and therefore, can lead to suboptimal decisions. Because of the two mentioned issues, a new configuration technology (*group-based configuration*) should be proposed to support a *community-based knowledge engineering*. This means a larger group of users (e.g., knowledge engineers, marketing experts, product developers, and sales representatives) should be integrated into knowledge engineering. Moreover, the new configuration mechanism should enable all users to be engaged in configuration processes and to jointly configure a product/service.

Concerning group-based configuration, very often, the preferences of an individual group member conflict with the configuration knowledge base and/or with the preferences of other group members. Such situations are referred to as *inconsistencies* which cause the *"no solution could be found"* dilemma (Reiterer et al., 2015). In single-user configurations, inconsistencies can be manually resolved by showing the conflicting preferences to the user and then let him/her decide which preference adaptations should be proceeded. In such scenarios, *minimal conflict sets* (Junker, 2004) are determined and the user manually performs a conflict resolution. Minimal conflict sets are employed to identify corresponding minimal diagnoses which have to be deleted/adapted from the user's preferences so that a solution can be found. However, in group-based configuration, it is more likely that many inconsistencies can be triggered among group members, and therefore, alternative diagnoses could be found. Therefore, a question arising in this context is *"which of the alternative diagnoses should be recommended first to the group?"*. Research to date has not yet answered this question, and to some extent, the resolution of inconsistencies in the context of group-based configuration has remained unclear.

The gaps mentioned above bring us to the first two research questions:

($Q1.1$) *How to support configuration for groups?*

($Q1.2$) *How to resolve inconsistencies in group-based configuration scenarios?*

2. **Improve preference acquisition processes in group-based configuration**

As mentioned in *Section 1.1*, configuration techniques (Stumptner, 1997; Felfernig et al., 2014a) are usually used by users to configure *complex items* which are characterized by many dimensions (attributes), such as *tourism packages* (Tran et al., 2016), *furniture* (Haag, 1998), *financial services* (Stolze et al., 2000; Jannach et al., 2010), and *release plans* (Ninaus et al., 2014). In the context of group-based configuration scenarios, each group member has to articulate his/her preferences concerning the dimensions of items. However, it could be the case that some group members are unable to evaluate the items because of the insufficient knowledge of group members about the domain or the lack of unshared (relevant) information, which is crucial for making optimal group decisions (i.e., *hidden profiles* (Greitemeyer and Schulz-Hardt, 2003)). Such an issue leads to *inaccurate evaluations* for items and consequently causes *inconsistencies* between group members' preferences. To avoid the negative impacts of knowledge gaps in the preference acquisition process, group members have to look for necessary information and then carefully analyze the items before articulating their preferences. However, these activities usually consume too much time and effort, and sometimes group members even cannot find helpful information to precisely evaluate the items.

This issue brings us to the third research question:

($Q2$) *How to better detect hidden profiles of group members in group-based configuration?*

3. **Provide explanations for group recommendations**

   When interacting with group recommender systems, users usually face troubles regarding *trust, effectiveness,* and *reliability* of group recommendations (Wang, 2016). The reason is that these systems often send group recommendations to users in the form of *"black-boxes"* where no explanation of the underlying mechanism of the recommendation process is shown. Therefore, the inclusion of explanations in group recommender systems can better help users evaluate the appropriateness of how their preferences have been considered in the group recommendation process. Moreover, the explanations could also help to increase the perception of group members concerning specific aspects of group decision making, such as *fairness* (i.e., take into account as far as possible group members' preferences) and *consensus* (i.e., help the group agree on a decision) (Felfernig et al., 2018d). Up to now, although extensive research has been carried out on explanations, most of them focus on explaining recommended items for single users (Herlocker et al., 2000; Sormo et al., 2005; Gkika and Lekakos, 2014; Lamche et al., 2014; He et al., 2016), whereas far too little attention has been paid to explaining recommendations for groups. In the current literature, there exist only a few research contributions with an in-depth analysis of explanations for group recommendations. For instance, Kapcak et al. (2018) and Najafian and Tintarev (2018) proposed explanations to describe underlying preference aggregation strategies and investigated how these explanations can help to improve group members' satisfaction with regard to group recommendations. However, these two approaches can only be applied to a certain scenario (e.g., sequential recommendations). Besides, they have not taken into account the fairness and consensus aspects among group members.

   The aforementioned gaps bring us to the fourth and the fifth research question:

   (*Q3.1*) *How to explain recommendations for groups?*

   (*Q3.2*) *How can explanations help to increase the fairness and consensus perception of users with regard to group recommendations?*

4. **Counteract decision manipulation in group recommender systems**

   As mentioned in *Section 1.1*, group recommender systems implementing user control mechanisms (McNee et al., 2003; Xiao and Benbasat, 2007; Jannach et al., 2017) could trigger decision manipulation issues. These mechanisms facilitate group members' rating adaptations, which cause decision manipulation (Jameson et al., 2003; Conitzer and Yokoo, 2010). Up to now, to some extent, it is still unclear how to counteract decision manipulation in group recommender systems. Although there are a couple of studies on manipulation resistance (Chirita et al., 2005; Van Roy and Yan, 2009; Li and Luo, 2011; Gunes et al., 2014; Zhou et al., 2015), most proposed solutions are applied to single-user recommendation scenarios. To the best of our knowledge, only a few research papers have attempted to resolve decision manipulation in group recommender systems. For instance, Stettinger et al. (2015) proposed a solution in which group members' preferences were not shown in the preference articulation phase. Jameson et al. (2003) developed a group recommender system applying a non-manipulatable aggregation strategy, such as *random strategy* (i.e., randomly selects an item from a given set of items) or *median strategy* (i.e., takes the item falling exactly in the middle of an ordered list of group members' preferences). Conitzer and Yokoo (2010) proposed a *mechanism design-based approach* to generate aggregation functions so that desirable recommendations can be achieved for groups, even if group members rated items based on their self-interests. All the mentioned approaches seem to be suboptimal solutions since they face the unacceptability of users concerning group recommendations (Jameson et al., 2003) or the obstacles in providing understandable explanations of group recommendations (Conitzer and Yokoo, 2010). Therefore, seeking for solutions that counteract decision manipulation in group recommender systems and meanwhile, guarantee optimal and explainable solutions for groups, has remained an open issue.

The mentioned issue brings us to the sixth research question:

(*Q4*) *How to counteract decision manipulation in group recommender systems?*

5. **An example application of recommender systems**

Recommender systems are usually used in online stores to recommend to users products/services that meet their needs and desires. In recent years, researchers have paid their attention to recommender systems in healthcare domains (Freyne et al., 2011; El-Dosuky et al., 2012; Elahi et al., 2015; Schäfer et al., 2017; Erdeniz et al., 2018). These systems not only provide users with personalized suggestions about *diagnoses, treatment methods,* and *diet* but also convince them to follow healthier lifestyles. In order to present useful and reliable suggestions to users, health recommender systems have to take into account users' preferences as well as their *personal health records* (Wiesner and Pfeifer, 2014). For instance, in the healthy food domain, besides users' preferences for food or recipes, recommender systems further consider users' *workout routines, nutrient regimens,* and *health problems* to recommend diets which suit their interests and health conditions. Up to now, proposing recommendation approaches to health recommender systems is still a new and upcoming field of research.

In this thesis, we choose the *healthy food domain* as the representative of healthcare domains. Along with the increase of *lifestyle-related illness* which are the cause of many *chronic diseases* such as diabetes or obesity, developing food recommender systems has become more urgent nowadays to propose appropriate dietary to users and to help them nourish themselves more healthily (Elsweiler et al., 2015). Although there exist a couple of studies on recommendation algorithms in the healthy food domain, most of them target at single-user recommendation scenarios. Group scenarios that are especially popular in the food domain (e.g., a group of friends decides on the menu of the party next week) have been paid very little attention. Up to now, in-depth discussions on the recommendation algorithms for groups in the healthy food domain is still an open topic. (Berkovsky and Freyne, 2010) is one of the studies which proposes some preference aggregation strategies to generate food recommendations for groups of users. However, the proposed solution has not considered aspects which are relevant to group recommendation scenarios, such as *"how to help group members achieve a consensus"* or *"how foster the fairness among group members"*.

On the other hand, existing studies in food recommender systems have been facing some research challenges that should be taken into account in future work. In these systems, information related to *user profiles* and *recipe descriptions* serves as a basis to generate recommendations. We argue that the *accuracy* and *adequacy* of this information could strongly influence the quality of recommendations. However, the current research literature on food recommender systems has not provided detailed discussions on this issue. Besides, how to convince users to change their eating behaviors and to follow healthy lifestyles is another challenge. In this context, adding explanations to food recommender systems could help to significantly increase the acceptance of users concerning recommended items. Although some papers (see, e.g., (Ueta et al., 2011) and (Aberg, 2006)) present some recommendation algorithms to tackle health issues, the persuasiveness of food recommendations are not considered.

The mentioned gaps motivate us to come up with the following two research questions:

(*Q5.1*) *How to generate food recommendations to groups of users?*

(*Q5.2*) *Which open issues in the healthy food domain should be taken into account within the scope of future work?*

6. **Further issues in group decision making**

Group recommendations are usually generated by aggregating the preferences of individual group members based on aggregation strategies (Masthoff, 2011). The outcomes of a strategy reflect the preferences of the whole group regarding a given set of items. The decision making behavior of group members could be influenced by different factors, especially by *item domains* (Felfernig et al., 2017). Item domains are categorized depending on how much decision effort that users invest for them. Basically, there are four types of item domains: *very low-, low-, high-,* and *very high- item domains* which respectively include items with *very low, low, high,* and *very high related decision efforts*. Thereby, when merging the preferences of group members, aggregation strategies should be chosen according to the item domains (Felfernig et al., 2017).

Besides item domains, the decision making behavior of group members can be sometimes influenced by the order of decision tasks, especially for scenarios in which a group of users has to continuously make decisions on a sequence of decision tasks from different item domains. For instance, family members are traveling together by car to some place for a couple of hours. Taking advantage of this occasion, they want to make different decisions in the following item domains: (1) *music genre* (e.g., selecting a music genre to be played while they are together in the car), (2) *restaurants* (e.g., choosing a restaurant to have dinner together next week), (3) *travel destinations* (e.g., selecting a tourism destination for the upcoming summer holiday), and (4) *apartment* (e.g., deciding on a new apartment to move in next month). In such a scenario, the mentioned item domains (1, 2, 3, 4) respectively indicate *very low-, low-, high-, and very high*-involvement item domains. In the beginning, if the group comes up with decision tasks (1) and (2), then a music genre/restaurant can be chosen using the *majority* strategy (i.e., choosing a music genre/restaurant that fits the preferences of the majority of group members). In contrast, if the group first completes decision tasks (3) and (4), then the strategies used in these tasks could have an impact on decision tasks (1) and (2). For instance, the music genre/restaurant could be selected based on the preferences of *all* (instead of *the majority*) group members since this strategy was earlier applied to decision tasks (3) and (4). These biases can be referred to as *serial position effects* in which the order of decision tasks could cause different decision making behaviors of group members. To the best of our knowledge, up to now, there do not exist any studies which give an in-depth analysis of how serial position effects influence the decision making behavior of group members in sequential group decision making. The investigation of related biases is the precondition of improving the prediction quality of group recommender systems.

The mentioned issue motivates us to come up with the last research question:

(*Q6*) *How do serial position effects influence the decision making behavior of group members in the context of sequential group decision making?*

1.3. Thesis Contributions

This thesis attempts to answer the research questions proposed in *Section 1.2*. The main contributions of the thesis with regard to the posed research questions are summarized in Table 1.1:

Research Questions	**Contributions**
(**Q1.1**) *How to support configuration for groups?*	We discuss a new configuration approach denoted as *group-based configuration* supporting scenarios in which groups of users jointly configure products or services (Felfernig et al., 2016).
(**Q1.2**) *How to resolve inconsistencies in group-based configuration scenarios?*	We introduce formal definitions to represent a group-based configuration task as a *constraint satisfaction problem* and show how inconsistencies between group members' preferences can be resolved based on *model-based diagnosis* (Stumptner, 1997). Besides, we also propose an approach to integrating aggregation heuristics (Masthoff, 2011; Felfernig et al., 2018a) into the diagnosis selection process to choose a minimal diagnosis which can be shown first to the group. This approach can be helpful in situations where many different diagnoses are found, and group members have no idea about which of these diagnoses should be chosen to solve the *"no solution could be found"* dilemma (Felfernig et al., 2016).
(**Q2**) *How to better detect hidden profiles of group members in group-based configuration?*	We propose a new approach to resolving group members' knowledge gaps in the context of group-based configuration. The main contribution lies in the application of the *liquid democracy* concept to improve the quality of the preference acquisition process in group-based settings. Liquid democracy (Zhang and Zhou, 2017) allows group members who lack of knowledge about *configuration items* (i.e., items characterized by many attributes) to delegate their votes to domain experts. In other words, the domain experts help the group members to evaluate items. Besides, we propose a novel approach to calculating the utility of configuration items based on *Multi-attribute Utility Theory (MAUT)-based evaluation* (Dyer, 2005) which emphasizes the role of the domain experts (Atas et al., 2018).

(**Q3.1**) *How to explain recommendations for groups?*	We provide an overview of explanations in single-user and group recommender systems. We propose possibilities of explaining recommendations in the context of basic recommendation techniques (e.g., collaborative filtering, content-based filtering, constraint-based, and critiquing-based recommendation) while considering specific aspects in group decision making. Moreover, some working examples of verbal explanations and corresponding visualizations are also provided to support a more in-depth understanding of the proposed approaches (Felfernig et al., 2018d).
(**Q3.2**) *How can explanations help to increase the fairness and consensus perception of users with regard to group recommendations?*	We propose different explanation types using *social-choice based preference aggregation strategies* (Masthoff, 2011; Felfernig et al., 2018a). These explanations intuitively explain the underlying mechanism of preference aggregation strategies and take into account *fairness* and *consensus* aspects among group members. Besides, we investigate explanations which best help to increase the fairness and consensus perception of users concerning group recommendations. Especially, we discover the *relationships* between the perceived fairness/perceived consensus of explanations and the satisfaction of users with regard to group recommendations. Moreover, in the context of repeated decisions, we also find out that taking into account the satisfaction of group members from the previous decisions helps to increase the fairness perception of users concerning group recommendations (Tran et al., 2019a).

(**Q4**) *How to counteract decision manipulation in group recommender systems?*	One of the user control mechanisms, which can cause decision manipulation in group recommender systems, is to allow users to adapt their preferences for items. Our contribution here is to propose user interfaces that visualize information dimensions regarding rating adaptations of group members at different transparency levels. The dimensions regarding group members' rating adaptations could be *"group members"* who have adapted the item ratings, *"items"* whose ratings have been adapted, *"ratings"* of items before and after being adapted, *"tendency"* which shows how the item ratings have been adapted, and *"changes of group recommendations"* after rating adaptations. We show that the user interface at the highest transparency level best helps to discourage users from decision manipulation. Besides, we find out that the ability of user interfaces to counteract decision manipulation differs depending on the dimensions represented in the user interface. The dimensions regarding *item ratings* and *group recommendations* have the strongest impact on preventing users from decision manipulation (Tran et al., 2019b).
(**Q5.1**) *How to generate food recommendations to groups of users?*	We provide an overview of how recommendations can be generated in the healthy food domain. Up to now, food recommender systems is still a growing field of research and has recently received much attention from researchers. Our contribution is to give insights into existing studies on food recommender systems and to show how well those systems can help users to choose recipes/food which meet(s) their tastes as well as their health conditions. Especially, we also discuss some approaches to generating food/recipe recommendations to groups based on the preferences of individual group members (Tran et al., 2018a).

(**Q5.2**) *Which open issues in the healthy food domain should be taken into account within the scope of future work?*	On the basis of analyzing existing studies on food recommender systems, we discuss some open issues that should be taken into account in future work: *(1)* collect *user-* and *food-related information* for the recommendation process; *(2)* improve the quality of recommendations; *(3)* explain recommended items in such a way that increases the trustworthiness and the persuasiveness of food recommendations; *(4)* apply psychological theories to drive users to healthy eating habits; *(5)* generate bundle recommendations based on negotiation and argumentation mechanisms; and *(6)* achieve a fast consensus in group decision making. Besides, we propose potential solutions to effectively consider the mentioned issues.
(**Q6**) *How do serial position effects influence the decision making behavior of group members in the context of sequential group decision making?*	We show the existence of *serial position effects* in a sequence of decision tasks from different item domains. We analyze different aggregation strategies (Masthoff, 2011; Felfernig et al., 2018a) to figure out which strategy is employed by group members in which sequence of decision tasks. Furthermore, we find out that the time invested in a group decision making process differs depending on the order of decision tasks in the sequence (Tran et al., 2018b).

Table 1.1.: Thesis contributions corresponding to the research questions.

1.4. Thesis Outline

The remainder of the thesis is structured as follows.

In **Chapter 2**, we first introduce formal definitions concerning a group-based configuration task and a corresponding solution. Thereafter, we show how inconsistent situations in group-based configuration can be resolved to achieve consensus within the group.

Chapter 3 describes a group-based configuration scenario in the context of requirements engineering, which is used as a working example throughout the chapter. Based on this scenario, we discuss a new group-based configuration approach in which the concept of *liquid democracy* is applied to transfer the rating task from group members to domain experts. Besides, this chapter presents a MAUT-based evaluation approach which calculates the utility values of configurable items on the basis of emphasizing the role of domain experts.

Chapter 4 provides an overview of existing studies on explanations in recommender systems for single users and groups. The chapter first presents primary goals and some example verbal explanations that are developed for single-user recommender systems. Thereafter, different approaches to generating explanations for groups are introduced in the context of collaborative filtering, content-based filtering, constraint-based, and critiquing-based recommendations. Besides, this chapter discusses social aspects within group recommendation scenarios and presents alternative approaches to visualize group recommendations.

Chapter 5 presents different types of textual explanations and shows which of these explanations best helps to increase certain social aspects in group decision making, such as the *fairness perception, consensus perception*, and *satisfaction* of group members with regard to group recommendations. In this chapter, we first present some *social choice-based preference aggregation strategies* and then show how the explanations for group recommendations can be formulated using these strategies. Thereafter, we propose some hypotheses and describe the main steps of a user study which has been conducted to examine these hypotheses. Finally, we present a summary of data analysis results and the indication of the explanation which best helps to improve the mentioned social aspects in group decision making.

Chapter 6 introduces user interfaces which can help to effectively counteract decision manipulation in group recommender systems. These user interfaces visualize the rating adaptation history of group members at different transparency levels. In this chapter, we first propose different dimensions that can be used to describe group members' rating adaptation history and then show how the user interfaces are visualized based on the proposed dimensions. Next, we define research questions and present the main steps of our user study which has been conducted to address the research questions. Finally, by analyzing the collected data from our user study, we show user interfaces which effectively help to discourage users from decision manipulation in group recommender systems.

Chapter 7 provides a summary of recommendation approaches that have been applied in the healthy food domain. In this chapter, we first present a short overview of basic recommendation techniques for single users. By analyzing existing studies, we categorize food recommender systems for single users based on the information that has been used in the recommendation process, such as *user preferences*, *nutritional needs*, *health conditions*, and *eating behaviors*. Next, we discuss a group decision making scenario in the food domain and show how a group recommendation can be created in such a scenario. Finally, we discuss research challenges that food recommender systems have been facing and some potential possibilities to take into account them within the scope of future work.

In order to further analyze the influence of cognitive biases on group decision making, in **Chapter 8**, we investigate the existence of the *serial position effects* in the sequential group decision making. Based on the results of our user study, we show that the decision making behavior of group members for a specific decision task differs depending on its position in a given sequence of decision tasks.

In **Chapter 9**, we conclude the thesis and present open issues for future work.

Chapter 2

Towards Group-based Configuration

This chapter is based on the results documented in (Felfernig et al., 2016). Major parts of this chapter in terms of writing and literature research are provided by the author of this thesis. For this work, we won the runner-up best paper award in the 18th International Configuration Workshop 2016 (ConfWS'16).

2.1. Abstract

Group-based configuration is a new configuration approach that supports scenarios in which a group of users is in charge of configuring a product or service. In this chapter, we introduce a definition of a group-based configuration task and a corresponding solution. Furthermore, we show how inconsistent situations in group-based configuration can be resolved to achieve consensus within the group. We introduce these concepts on the basis of a working example from the domain of (group-based) software release planning.

2.2. Introduction

Configuration (Stumptner, 1997; Felfernig et al., 2014a) is considered as one of the most successful applications of Artificial Intelligence technologies. It is applied in many domains such as *financial services* (Felfernig et al., 2007c), *telecommunication* (Fleischanderl et al., 1998), and the *furniture industry* (Haag, 1998). Configuration environments are typically *single-user oriented*, i.e., the underlying assumption is that a specific user is in charge of completing the configuration task. However, considering configuration as a single-user task can lead to suboptimal decisions (Felfernig et al., 2012b). For example, release planning is a task that typically requires the engagement of a group of stakeholders where the knowledge and preferences of all stakeholders should be taken into account to be able to achieve high-quality decisions (Greitemeyer and Schulz-Hardt, 2003; Felfernig et al., 2012b).

There are various scenarios where configuration decisions are not taken by a single person but by a group of users (Felfernig et al., 2014b). As mentioned, *Software Release Planning* (Felfernig et al., 2012b) is a requirements engineering related task, where groups of users (stakeholders) are deciding about the ordering in which requirements should be implemented. In this scenario, stakeholders have different preferences and knowledge regarding the implementation alternatives. Consequently, requirements related knowledge should be exchanged as much as possible, and existing contradictions in preferences and evaluations have to be resolved. *Holiday Planning* (Jameson et al., 2004) is another scenario where a group is in charge of identifying a configuration that is accepted by all group members - examples of related decisions are *region to visit, hotel,* and *activities* during the stay. *Product Line Scoping* (Schmid, 2000) is related to the task of determining boundaries in a product line. This task is a specific type of requirements engineering task and related decisions are crucial for the success of a whole product line effort. *Investment Decisions* (e.g., project funding) (Felfernig et al., 2014b) are often taken by a group of users who have to take into account constraints with regard to the overall amount of money that can be invested and the topics projects should deal with. The overall configuration task in this context is to identify a bundle of project proposals that take into account the financial limits and includes high-quality proposals.

Existing configuration environments do not take into account the aspect of group configuration (Felfernig et al., 2014b). In contrast, for *non-configurable* items such as movies, restaurants, personnel decisions, and music, there already exist proposals on how to support related group decision processes (Masthoff, 2004, 2011; Stettinger, 2014). In this context, *group recommendation heuristics* (Masthoff, 2011) are applied to support groups in their decision making activities. In order to achieve consensus, different decision heuristics are applied, which propose decisions acceptable for a group as a whole. For example, the *least misery* heuristic proposes alternatives which do not represent an absolute no-go for at least one of the group members. Besides decision heuristics, *standard recommendation approaches* (Jannach et al., 2010) such as *matrix factorization* can be applied to predict recommendations acceptable for a group as whole. These approaches rely on existing group recommendations. Based on such information about group selection behavior, corresponding recommendations can be determined for similar groups.

In this chapter, we focus on introducing a formal definition of a group configuration problem and show how inconsistencies in the preferences of group members can be resolved.* The remainder of the chapter is organized as follows. In *Section 2.3*, we introduce a basic definition of a group-based configuration task and introduce a corresponding example configuration knowledge base. In *Section 2.4*, we discuss approaches that can help to resolve inconsistencies in the preferences of individual group members. Finally, we discuss further issues for future work in *Section 2.5* and conclude the chapter in *Section 2.6*.

2.3. Group-based Configuration

In the following, we introduce definitions of a group configuration task and a corresponding solution. These definitions are based on a *Constraint Satisfaction Problem* (CSP) (Tsang, 1993) which is frequently used for the definition of single-user configuration tasks. The major characteristic of group-based configuration compared to other types of group decision tasks is that the alternatives are defined in terms of a knowledge base, i.e., the alternatives are not pre-specified. This requires new approaches to configuration and diagnosis search, and to represent the configuration task in a corresponding user interface.

Definition 1: *Group-based Configuration Task*. A group-based configuration task can be defined as a $CSP(V,D,C)$ where V is a set of variables, D represents the corresponding domain definitions, and $C = PREF \cup CKB$ represents a set of constraints. In this context, $PREF = \bigcup PREF_i$ is the union of customer preferences $PREF_i$ and CKB represents a configuration knowledge base.†

Definition 2: *Group-based Configuration*. A group-based configuration (solution) for a group-based configuration task is a complete set of assignments $CONF = \bigcup a_i : v_i = v_{ai}$ to the variables $v_i \in V$ such that $CONF \cup PREF \cup CKB$ is consistent.

Example 1: *Group-based Configuration Task*. For demonstration purposes, we introduce a simplified group-based configuration task from the domain of *software release planning*. The goal of software re-

*The work presented in this chapter has been developed within the scope of the WEWANT project (Enabling Technologies for Group-based Configuration) which is funded by the Austrian Research Promotion Agency (850702).

†We denote customer requirements as preferences (*PREFS*) in order to distinguish these from *software requirements* in the working example.

lease planning is to assign a corresponding release to each software requirement. In this example, nine requirements are represented in terms of variables $V = req_1, req_2, .., req_9$ and releases are represented as variable domains. If we assume that *three releases* have been planned for completing the whole software (i.e., implementing each individual requirement), each variable has a corresponding domain [1..3], e.g., $dom(r_1) = [1..3]$. For this example, we assume the existence of three stakeholders who are in charge of release planning - $PREF_i$ represents the preferences of stakeholder i.

- $V = \{req_1, .., req_9\}$
- $D = \{dom(req_1) = [1..3], .., dom(req_9) = [1..3]\}$
- $PREF_1 = \{pref_{11} : req_1 = 1, pref_{12} : req_2 = 1, pref_{13} : req_3 = 1, pref_{14} : req_5 = 2, pref_{15} : req_8 = 3\}$
- $PREF_2 = \{pref_{21} : req_3 = 1, pref_{22} : req_4 = 2, pref_{23} : req_6 = 3, pref_{24} : req_7 = 3\}$
- $PREF_3 = \{pref_{31} : req_5 = 2, pref_{32} : req_6 = 3, pref_{33} : req_8 = 3, pref_{34} : req_9 = 2\}$
- $CKB = \{c_1 : req_1 < req_5, c_2 : req_2 < req_8, c_3 : req_3 < req_6, c_4 : req_3 \neq req_4\}$

Example 2: *Group-based Configuration*. On the basis of the example group-based configuration task, a constraint solver could determine the following solution:

$CONF = \{a_1 : req_1 = 1, a_2 : req_2 = 1, a_3 : req_3 = 1, a_4 : req_4 = 2, a_5 : req_5 = 2, a_6 : req_6 = 3, a_7 : req_7 = 3, a_8 : req_8 = 3, a_9 : req_9 = 2\}$.

For each requirement, the constraint solver proposes a corresponding release in the context of which the requirement should be implemented.

2.4. Resolving Inconsistencies in Group Preferences

In the example introduced in *Section 2.3*, the basic assumption is that the preferences of individual group members are consistent. However, in group-based configuration scenarios, it happens quite often that the preferences of individual users differ. In the context of release planning scenarios, it is often the case that stakeholders have different preferences regarding the implementation of specific requirements. One requirement could be favored since the stakeholder needs the corresponding functionalities. Another reason could be that a stakeholder has no preferences or does not understand the requirement in detail. Inconsistencies between preferences can be manually resolved by showing inconsistent preferences to stakeholders and let them decide which changes should be performed. In such scenarios, minimal conflict sets are determined (Junker, 2004) and users in a manual fashion perform conflict resolution.

Alternatively, conflicts between requirements can be resolved automatically by calculating minimal diagnoses (*Definition 4*) for minimal conflict sets (*Definition 3*).

Definition 3: *Conflict Set*. A conflict set $CS \subseteq \bigcup REQ_i$ is a minimal set of requirements such that inconsistent(CS). CS is minimal if there does not exist a conflict set CS' with CS' is a conflict set and $CS' \subset CS$.

Minimal conflict sets can be exploited for determining the corresponding diagnoses (Reiter, 1987). Assuming that $\bigcup PREF_i \cup CKB$ is inconsistent, a minimal diagnosis (*Definition 4*) represents a minimal set of requirements that have to be deleted from $\bigcup PREF_i$ such that a solution can be found for the remaining

Stake-holder	req_1	req_2	req_3	req_4	req_5	req_6	req_7	req_8	req_9
1	$pref_{11}$: $req_1 = 2$	$pref_{12}$: $req_2 = 1$	$pref_{13}$: $req_3 = 1$		$pref_{14}$: $req_5 = 2$			$pref_{15}$: $req_8 = 3$	
2			$pref_{21}$: $req_3 = 2$	$pref_{22}$: $req_4 = 3$		$pref_{23}$: $req_6 = 3$	$pref_{24}$: $req_7 = 3$		
3					$pref_{31}$: $req_5 = 2$	$pref_{32}$: $req_6 = 3$		$pref_{33}$: $req_8 = 3$	$pref_{34}$: $req_9 = 2$

Table 2.1.: Tabular representation of constraints in a group-based configuration task (*Example 3*). Conflict set $CS_1 = \{pref_{11}, pref_{12}\}$ reflects inconsistent preferences of stakeholder 1 (the preferences are inconsistent with the configuration knowledge base) and conflict set $CS_2 = \{pref_{13}, pref_{21}\}$ reflects a conflict between the preferences of stakeholders 1 and 2.

constraints (see *Definition 4*).

Definition 4: *Group-based Configuration Diagnosis Task.* A group-based configuration diagnosis task is defined by a group-based configuration task $(V, D, C = PREF \cup CKB)$ where $PREF \cup CKB$ is inconsistent.

Definition 5: *Group-based Configuration Diagnosis.* A diagnosis for a given group-based configuration task $(V, D, C = PREF \cup CKB)$ is a set Δ such that $CKB \cup PREF - \Delta$ is consistent. Δ is minimal if $\neg\exists\Delta' : \Delta' \subseteq \Delta$.

Example 3: *Group-based Configuration Diagnosis Task.* An example group-based configuration task that includes inconsistencies between different user requirements is the following.

- $V = \{req_1, .., req_9\}$
- $D = \{dom(req_1) = [1..3], .., dom(req_9) = [1..3]\}$
- $PREF_1 = \{pref_{11} : req_1 = 2, pref_{12} : req_2 = 1, pref_{13} : req_3 = 1, pref_{14} : req_5 = 2, pref_{15} : req_8 = 3\}$
- $PREF_2 = \{pref_{21} : req_3 = 2, pref_{22} : req_4 = 3, pref_{23} : req_6 = 3, pref_{24} : req_7 = 3\}$
- $PREF_3 = \{pref_{31} : req_5 = 2, pref_{32} : req_6 = 3, pref_{33} : req_8 = 3, pref_{34} : req_9 = 2\}$
- $CKB = \{c_1 : req_2 > req_1, c_2 : req_2 < req_8, c_3 : req_3 < req_6, c_4 : req_3 \neq req_4\}$

In this example, the requirements of the first stakeholder are inconsistent since the combination $req_1 = 2$ and $req_2 = 1$ is inconsistent with the underlying knowledge base ($c_1 : req_2 > req_1$). Furthermore, there exists an inconsistency between the requirements $req_3 = 1$ (*stakeholder 1*) and $req_3 = 2$ (*stakeholder 2*).

The minimal conflict sets that can be derived from our working example are the following: $CS1 = \{pref_{11}, pref_{12}\}$ and $CS2 = \{pref_{13}, pref_{21}\}$. The corresponding set of alternative *diagnoses* (also called *hitting sets*) is the following: $\Delta_1 = \{pref_{11}, pref_{13}\}$, $\Delta_2 = \{pref_{11}, pref_{21}\}$, $\Delta_3 = \{pref_{12}, pref_{13}\}$, and $\Delta_4 = \{pref_{12}, pref_{21}\}$. A diagnosis is a minimal set of requirements from $\bigcup PREF_i$ such that $CKB \cup PREF - \Delta$ is consistent.

Diagnoses represent a set of *consistency-preserving delete operations* that can be applied to the set $\bigcup PREF_i$ in the case that $PREF \cup CKB$ is inconsistent. In many cases, there exist different diagnoses that

Stakeholder	$\Delta_1 = \{r_{11}, r_{13}\}$	$\Delta_2 = \{r_{11}, r_{21}\}$	$\Delta_3 = \{r_{12}, r_{13}\}$	$\Delta_4 = \{r_{12}, r_{21}\}$
1	2	1	2	1
2	0	1	0	1
3	0	0	0	0

Table 2.2.: Overview of the impact of the different diagnoses Δ_i on the current preferences of stakeholders, for example, stakeholder 1 has to change two of his/her requirements if diagnosis Δ_1 gets selected.

Heuristic	$\Delta_1 = \{r_{11}, r_{13}\}$	$\Delta_2 = \{r_{11}, r_{21}\}$	$\Delta_3 = \{r_{12}, r_{13}\}$	$\Delta_4 = \{r_{12}, r_{21}\}$
Least Misery	2.0	1.0	2.0	1.0
Average	0.67	0.67	0.67	0.67
Most Pleasure	0.0	0.0	0.0	0.0

Table 2.3.: Evaluation of the different diagnoses using the *least misery*, *average*, and the *most pleasure* heuristics. In all three heuristics, the ranking criteria for the diagnoses is *"less is better"*.

can be recommended for preserving the consistency between user requirements and the configuration knowledge base (*CKB*). A ranking of alternative diagnoses in the context of group configuration scenarios can be achieved, for example, by determining a candidate set of minimal diagnoses that is then ranked on the basis of different types of group decision heuristics (Masthoff, 2011).

An example of the application of such group decision heuristics will be discussed in the following. Table 2.1 depicts a situation where individual user requirements are inconsistent. In order to resolve this inconsistency, the alternative diagnoses $\Delta_1, \Delta_2, \Delta_3$, and Δ_4 can be applied. An open question in this context is *"which of the alternative diagnoses should be recommended first to the group of users"*. Table 2.2 summarizes the impact of the different diagnoses on the current preferences of stakeholders (users). For this purpose, different group decision heuristics can be applied that help to figure out alternatives acceptable for the whole group. In the following, we exemplify *three basic heuristics* and show how these can influence the selection of a diagnosis.

$$LeastMisery(\Delta) = argmax_d \bigcup_{s \in users} pref_\delta(s, \Delta) = d \tag{2.1}$$

$$Average(\Delta) = \frac{\sum_{s \in users} pref_\delta(s, \Delta)}{|users|} \tag{2.2}$$

$$MostPleasure(\Delta) = argmin_d \bigcup_{s \in users} pref_\delta(s, \Delta) = d \tag{2.3}$$

Least Misery: For each diagnosis, we first calculate the number of requirements that each group member has to adapt. For instance, with the diagnosis Δ_1, *user 1* has to adapt two requirements (r_{11} and r_{13}), whereas *user 2* and *user 3* do not have to adapt any requirement (see Table 2.2). Thereafter, Formula 2.1 is applied to evaluate the least misery values of the diagnoses (see Table 2.3). Finally, the diagnosis with the *minimum least misery value* will be recommended first to the whole group. In this context, the least misery value for a whole group is the minimum of the maximum number of preferences part of a diagnosis, i.e., *the lower the least misery value, the better the corresponding diagnosis*. In our example, either Δ_2 or Δ_4 will be recommended first to the group since both of them achieve the lowest numbers of adaptations.

Average: Based on Table 2.2, we can evaluate the average value of each diagnosis using Formula 2.2. Thereafter, the diagnosis with the *lowest average value* will be recommended to the group. In our example, the average value of each diagnoses Δ_i is 0.67 and therefore any of them can be recommended to the group.

Most Pleasure: This heuristic prefers the diagnosis with the best outcome for one user (see Formula 2.3). For example, in Table 2.3, the most pleasure value of each diagnosis Δ_i is 0.0 since for *user 3*, there does not exist a need to adapt his/her preferences in all of the diagnoses. Therefore, with the most pleasure approach, we can choose any diagnosis to recommend to the group.

2.5. Future Work

The primary goal of this chapter is to present our initial ideas related to the implementation of group-based configuration technologies. There are a couple of issues to be solved within the scope of future work. These issues will be discussed in the following paragraphs.

Consensus in Group Decision Making. Presenting diagnoses in situations where user preferences are inconsistent with the underlying configuration knowledge base and/or the preferences of other users is a basic means to trigger discussions and achieve consensus (Felfernig et al., 2012b). However, further aspects have to be taken into account to accelerate the achievement of consensus in group decision making. Promising approaches in this context are, for example, in the following. User interfaces have to be enriched in order to allow basic negotiation mechanisms between users. An example thereof is as follows: *stakeholder A* is interested in having implemented requirement req_a as soon as possible. Furthermore, *stakeholder B* is interested in having implemented requirement req_b as soon as possible. *Stakeholder A* would accept an earlier implementation of req_b if *stakeholder B* accepts an earlier implementation of requirement req_a. In this context, visualization concepts for the representation of the current decision situation will play a crucial role - alternative ways to represent decision situations are a focus of future work.

Fairness in Group Decision Making. An important issue in group decision making is the fairness of group members. Fairness is a primary topic within the scope of repeated decision processes where the same or similar groups are taking a decision. A related example is holiday decisions where a group of friends decides about a new travel destination and related activities. The preferences of users who were discriminated to some extent in the previous years travel arrangements should have a higher emphasis on the new holiday decision. Fairness also includes visualization aspects since the visualization of the current state of the decision process could help to increase fairness in group decision making, for example, by increasingly taking into account the preferences of other group members.

Predictive Search. Based on the information of completed group decision processes, diagnosis and repair could be improved by better predicting alternatives acceptable for the whole group. In this context, different types of personalization approaches should be included that help to take into account the preferences of the whole group when determining diagnoses and corresponding repair actions. Diagnosis prediction approaches for single users are already discussed in related work (Felfernig et al., 2014a). However, in group decision scenarios, further related aspects have to be taken into account. The prediction of a relevant diagnosis does not only have to take into account the selection behavior of users but also how users interacted with each other within the scope of a group decision process. Furthermore, the search for alternative configurations has to consider group preferences, i.e., search heuristics must be learned on the

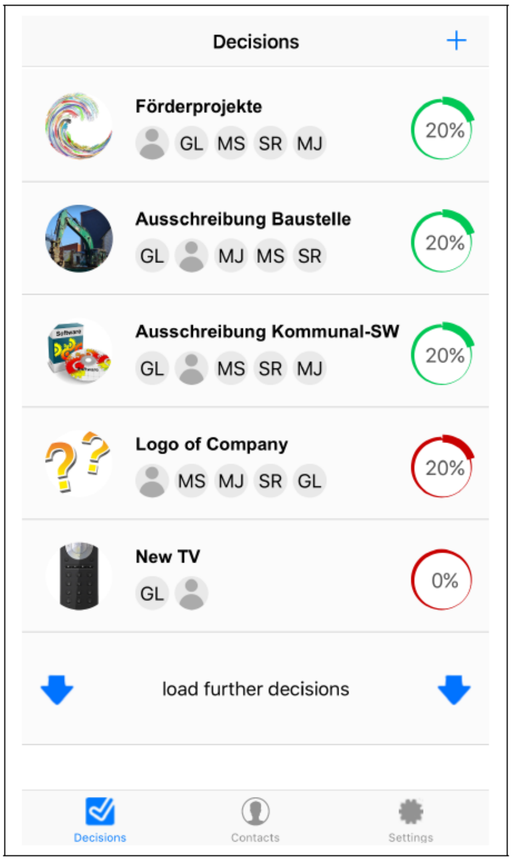


Figure 2.1.: CHOICLA group decision support environment. Each entry represents a group decision task - the corresponding percentages indicate the share of users who already articulated their requirements. A red circle indicates the fact that the current user did not articulate his/her preferences.

basis of past group interactions.

Negotiation Mechanisms. The main challenge of negotiation mechanisms is to integrate these into configuration systems in such a way that is easy to understand for users. End-users will not accept complex negotiation mechanisms; therefore the major challenge is to propose decision and negotiation mechanisms that help to achieve high-quality decisions and consensus as soon as possible and to trigger inconsistency management only in situations where real disagreements exist. For example, if one stakeholder evaluates the risk level of a requirement with 7 (on a scale of [1..10]) and the other stakeholder evaluates the same requirement with 8, there seems to be no real disagreement and the system may not have to point out an existing inconsistency.

Intelligent User Interfaces. Since group-based configuration tasks are solved in a distributed and asynchronous fashion, user interfaces should be able to take into account this situation. *Figure 1* includes a screenshot of the CHOICLA group decision support environment (Stettinger, 2014).[‡] CHOICLA is an application that supports group decisions related to *non-configurable* products and services (e.g., party locations and type of dinner), i.e., decisions are taken with regard to a collected assortment of alternatives but are not taken with regard to certain attributes (variables) which are basic elements of a configuration task. The only possibility of CHOICLA to take decisions concerning configurable products is to enumerate a representative

[‡]The latest version of CHOICLA can be found at www.choiclaweb.com

set of alternatives (e.g., new family car). In future versions of CHOICLA, we will support the integration of complete configuration tasks into decision processes. Variables will then be represented as alternatives and user preferences, and inconsistencies will be represented on a corresponding graphical level.

2.6. Conclusion

In this chapter, we introduced the concept of group-based configuration. We also presented basic definitions of a group-based configuration task (represented as a constraint satisfaction problem) and showed how to deal with inconsistent preferences of group members based on the concepts of model-based diagnosis. In this context, we showed how to integrate different types of decision heuristics into diagnosis selection processes. Finally, we discussed different challenges for future work we want to tackle.

Chapter 3

Liquid Democracy in Group-based Configuration

The contents of this chapter are based on the results documented in (Atas et al., 2018). The author of this thesis provided major contributions regarding literature research, proposing solutions, and writing major parts of this chapter. For this work, we received the Best Paper Award in the 20th International Configuration Workshop (ConfWS'2018).

3.1. Abstract

Group-based configuration systems support scenarios where a group of users configures a product or service. In group-based configuration scenarios where the knowledge of some group members regarding items is insufficient, the advice of experts is necessary to help them evaluate products or services. This chapter introduces a novel approach which applies the concept of *liquid democracy* to delegate the voting power of group members to experts. Concerning the application of liquid democracy, we propose an approach to calculating the utility of configurable items based on *Multi-attribute Utility Theory (MAUT)-based evaluation*. Compared to the traditional approach, the proposed MAUT-based evaluation focuses on the role of experts by assigning a higher weight (importance) to them. Besides, the individual expertise level of experts is taken into account in the utility calculation of items. Consequently, the main contribution of this chapter consists of the improvement of group-based configuration by taking *liquid democracy* into consideration.

3.2. Introduction

Configuration (Stumptner, 1997; Felfernig et al., 2014a) is an important application area of Artificial Intelligence that enables users to configure *complex items* described by many dimensions (attributes). Typical examples of such items include *release plans* (Ninaus et al., 2014), *tourism packages* (Tran et al., 2016), *furniture* (Haag, 1998), and *financial services* (Stolze et al., 2000; Jannach et al., 2010). While most existing configuration systems focus on the support of single users, there also exist scenarios where groups of users can jointly configure items, for instance, *requirements engineering* scenarios where a group of stakeholders configures software release plans. In such scenarios, *group-based configuration systems* have been recognized as being useful tools that help to identify configurations which satisfy the preferences of all group members (Felfernig et al., 2016). When interacting with group-based configuration systems, each group member explicitly articulates his/her preference for different item dimensions. Preferences articulated by group members are then checked for consistency. As soon as all user preferences are consistent with each other as well as with the knowledge base, the constraint solver will be able to find items that satisfy the preferences of all group members. Thereafter, the utility value of each item can be calculated, for example, based on *Multi-attribute Utility Theory (MAUT)* (Dyer, 2005). This approach takes into account the preferences of group members concerning the dimensions of items and the importance of the dimensions from the users' point of view. The item achieving the highest utility value will then be recommended to the group.

In the context of group-based configuration, some group members might be sometimes *unable* to evaluate the dimensions of a given set of items due to a knowledge gap. Hence, to precisely evaluate items, group members have to invest much effort in collecting necessary information and then analyzing the items (Zhang and Zhou, 2017). In such a situation, group members could ask for advice from people who are experts in the item domain of interest. The consultations of experts help to precisely identify the items' evaluations and thereby further facilitates the entire configuration process. The preference configuration of group members in this context can be interpreted and considered as a *liquid democracy* paradigm, which provides an alternative decision making model to make better use of collective intelligence (Zhang and Zhou, 2017). The *liquid democracy* concept empowers group members to either play an *active role* (i.e., *active users* who directly vote items) or a *passive role* (i.e., *passive users* who delegate their rating power to experts) in the voting process (Boldi et al., 2015).

Recently, a variety of studies regarding liquid democracy have been conducted to make better use of the so-called *"wisdom of the crowds"* (Zhang and Zhou, 2017). For instance, Boldi et al. (2015) proposed a Facebook application that enables each user to select one of his/her friends as the expert of a music genre. The expert then helps him/her to select some pieces of music. Johann and Maalej (2015) applied liquid democracy and e-democracy concepts to address the challenges of massive and continuous user participation in the context of requirements engineering. Zhang and Zhou (2017) proposed an efficient *statement voting* scheme that unifies two basic stages of liquid democracy, i.e., *delegation* and *voting*. During the voting/delegating phase, each voter can either vote for the candidate(s) or delegate his/her voting power to another voter. Each voter is denoted by a *temporal ID* which is encrypted and distributed in such a way that guarantees the anonymity of the delegation/voting process.

Up to now, although there exist a couple of studies on liquid democracy, to some extent, it is still unclear how liquid democracy can be applied in the context of group-based configuration. Two emerging questions are: (i) *"How does the system recommend experts to a user who has not enough knowledge about items?"* and (ii) *"How to calculate the utility of an item on the basis of emphasizing the importance of experts who were chosen by stakeholders?"*. To the best of our knowledge, there does not exist any research which provides an in-depth view of the correct application of liquid democracy in the group-based configuration. In this chapter, we present an insight into the application of liquid democracy in the group-based configuration. Besides, we propose a novel approach of MAUT-based evaluation that takes the preferences of group members/experts into account and thereby assigns higher importance to the experts.

The remainder of the chapter is organized as follows. In *Section 3.3*, we describe a group-based configuration scenario in requirements engineering, which is used as a working example throughout the chapter. In *Section 3.4*, we discuss how liquid democracy can be applied to a group-based configuration in order to transfer the rating power from group members to experts. *Section 3.5* presents a new approach of MAUT-based evaluation to calculate the utility value of a requirement. *Section 3.6* discusses how requirements can be assigned to releases based on their utility values, their effort estimations, existing dependencies between requirements, and the capacity of releases. Finally, *Section 3.7* draws a brief conclusion and provides some ideas for future work.

Requirement	Title	Description
R_1	Evaluation Software	To evaluate the collected training data, an evaluation software is required. The evaluation software requires the connection and access to the clock's internal memory. The evaluation should contain measured information regarding the distance, the height, the average heart rate, and the calorie consumption.
R_2	Data-Storage Function	To evaluate the measured data, a storage service is required. The internal memory is used for saving the measured information, such as the distance, the height, the average heart rate, and the calorie consumption. The stored data will be used by the evaluation software.
R_3	GPS	To identify the position, a GPS sensor is used. Based on the measured position and time information, the speed and the distance can be measured.
R_4	Display lighting	The sport watch needs a display lighting to be operated at dusk.

Table 3.1.: Example requirements for the development of a *sport watch*. Each requirement is described by an *id*, a *title*, and a *textual description*.

3.3. Working Example

For demonstration purposes, we introduce a group configuration scenario occurring in a small requirements engineering example project where we configure a release plan. In this context, we define a set of requirements (R_1, R_2, R_3, and R_4) for developing a *sport watch*. These requirements are defined by a group of engineers with longstanding experience and practical knowledge in requirements engineering. Each requirement is described by an *id*, a *title*, and a *textual description* (see Table 3.1). In this example, we assume a situation where a group of five stakeholders (S_1, S_2, S_3, S_4, and S_5) read requirements, evaluate them according to different dimensions, and assign them to different releases (i.e., release planning configuration). We defined two different releases which are shown in Table 3.2.

Given the sets of requirements and releases, we assume that each stakeholder evaluated the requirements concerning the following dimensions: *risk, effort*, and *profit*. The *risk* indicates the estimated risk of developing a requirement. The *effort* represents the estimated total work done for developing a requirement. The *profit* corresponds to the estimated profit of a requirement. These dimensions are evaluated using ratings lying in the range between 1 and 5. An evaluation of 5 indicates the requirement with *low risk, high profit,* and *low effort*, whereas an evaluation of 1 represents the requirement with *high risk, low profit,* and *high effort*.

The evaluation of stakeholders for the requirements is shown in Table 3.3. In this table, some group members did not sufficiently evaluate the requirements (i.e., some dimensions of the requirements have not been evaluated). For instance, the first stakeholder (S_1) did not evaluate the *profit* of requirement R_2.

Release	Capacity *(in hours)*	Start date	End date
Release 1	260	2020-05-01	2020-07-01
Release 2	260	2020-07-15	2020-09-15

Table 3.2.: Defined releases for the development of a *sport watch*. Each release is described by the *start date*, the *end date,* and the *capacity*. The *capacity* indicates the planned effort of a release which is measured in hours.

Besides, there also exist some stakeholders who did not evaluate any dimension of a requirement. For instance, stakeholder (S_4) did not evaluate any dimension of requirement R_1. A possible reason for this missing can lie in the expertise or knowledge deficiency of the stakeholders regarding the requirements. Therefore, the stakeholders might invest much effort to acquire the necessary knowledge and to analyze the requirements. This triggers a high cost of the requirement evaluation process. In this scenario, the stakeholder could ask for the advice of some experts to provide more accurate evaluations. In other words, the stakeholder directly passes his/her evaluation power to experts by using *liquid democracy* (see *Section 3.4*). In requirements engineering, experts can be requirement engineers who have longstanding experiences and practical knowledge of requirements. The consultation of experts helps to precisely evaluate the dimensions of items.

On the other hand, in some cases, empty evaluations could be triggered by the fact that the stakeholder does not want to evaluate the dimensions of a requirement, and he/she does not want to delegate the rating power to anyone else. In this scenario, group-based configuration systems will automatically check the number of complete evaluations of the requirements, and the configuration phase is only complete if this number reaches a predefined threshold. In our example, we assume the number of complete evaluations should not be *lower than 80%* of the total number of all evaluations. In other words, the configuration phase will not finish until the number of available evaluations reaches 80%.

In addition to that, when evaluating a requirement, stakeholders can assign different *weights* to the dimensions. The weight refers to the *importance* of a dimension, which means *the higher the importance of a dimension, the higher the weight*. Different stakeholders could assign different weights to the same dimension. For instance, a software developer can assign the highest weight to the *effort* of a requirement, whereas a project manager might evaluate the *profit* of a requirement to be the most important dimension. In order to limit the scope of this chapter, some simplifications have to be made. For the sake of simplicity, in our working example, we assume the stakeholders assign the same weights to all dimensions of the requirements and they have the weight of 1 from the stakeholders' point of view (i.e., $\forall s \in$ stakeholders, *weight*(*s, risk*) = *weight*(*s, profit*) = *weight*(*s, effort*) = 1).

3.4. Application of Liquid Democracy

Liquid democracy is a hybrid voting model of participative democracy which combines *direct* and *representative democracy* approaches to empower electors (Litvinenko, 2012; Blum and Zuber, 2016). While *direct democracy* allows electors to directly vote for an item, *representative democracy* enables electors to choose representatives (or experts) and empower them to vote for items. One of the major issues of direct democracy is the insufficient knowledge of the voter about some items. As a result, this voter may provide inaccurate evaluations or even not be able to assess them reasonably. In sharp contrast to direct democracy, representative democracy allows a stakeholder to select an expert who plays the role of a representative to vote for items. However, representative democracy is also known to show a weakness in terms of *representativeness*. In particular, this is true for scenarios where many voters delegate their

Requirement		Stakeholder				
		S_1	S_2	S_3	S_4	S_5
R_1	*risk*	5	3	3	-	3
	profit	3	3	4	-	3
	effort	4	-	3	-	4
R_2	*risk*	2	2	5	4	-
	profit	-	-	-	-	-
	effort	4	3	4	2	4
R_3	*risk*	4	2	2	2	2
	profit	4	5	3	4	-
	effort	4	-	3	4	4
R_4	*risk*	2	2	4	1	4
	profit	-	5	2	3	3
	effort	-	4	-	-	4

Table 3.3.: Evaluations of stakeholders for the defined requirements in Table 3.1. Each requirement is represented by the following three dimensions: *risk, profit,* and *effort*. Each evaluation is in the range of 1 to 5. An evaluation of 5 indicates a requirement with *low risk, high profit,* and *low effort*. An evaluation of 1 represents a requirement with *high risk, low profit,* and *high effort*. Evaluations which were not provided by stakeholders are represented by dash symbols ('-').

Expert	Expertise level (*sport watch* domain)
$Expert_2$	4.5
$Expert_5$	3.75
$Expert_4$	3.15
$Expert_1$	2.25
$Expert_3$	2.05

Table 3.4.: The expertise level of the experts in the *sport watch* domain. The expertise level is in the range of 1 to 5, whereby 1 indicates limited knowledge and 5 indicates excellent knowledge.

voting power to only one expert. This means, the expert's opinion usually represents the idea of many voters. Hence, it triggers a situation where the evaluations of the expert only reflect the opinion of one or some voter(s), but not all of them. In this context, liquid democracy has been recognized as a mixed approach that takes advantage of the strength of direct and representative democracy. It enables voters to either directly vote items or delegate their voting rights to an expert. Consequently, this key benefit of liquid democracy serves as the main motivation to apply this voting model.

In this chapter, we use a liquid democracy approach in order to complete the evaluations of dimensions which were not evaluated by stakeholders. In our example, stakeholders S_2, S_4, and S_5 did not evaluate all dimensions of the requirements (see Table 3.3) and they need help from experts to complete their evaluations. The expert selection can be made by one of the following approaches. The first approach is to select *only one expert* for the stakeholders. The second approach is to allow each group member to select his/her expert. In our example, we choose the second approach in which each stakeholder chooses different experts for different requirements. For instance, regarding the requirement which is related to user interfaces, the stakeholder can choose an expert who has many experiences in user interface design. For the data storage-related requirement, the stakeholder can choose an expert who is knowledgeable about

Stakeholders	R_1	R_2	R_3	R_4
S_2	$Expert_3$	$Expert_2$	$Expert_4$	-
S_4	$Expert_2$	$Expert_2$	-	$Expert_2$
S_5	-	$Expert_5$	$Expert_2$	-

Table 3.5.: Experts chosen by stakeholders regarding different requirements. The dash symbol '-' represents a situation in which a stakeholder does not need any advice of an expert.

data management. In our approach, the expert selection process is done automatically by a recommender system. That means experts on a specific topic are automatically identified and recommended to the stakeholder. Alternatively, each stakeholder is allowed to select experts who are not included in the recommended list. In our approach, the recommender system suggests experts based on the *expertise level*. In the context of requirements engineering, the expertise level of an expert can be calculated based on the following criteria: *working experience*, *skills*, *number of contributions* in requirements engineering projects, and *number of delegations* received in the requirements engineering domain. The expertise level is in the range of 1 to 5, whereby 5 indicates an *excellent* topic-related knowledge and 1 represents a *limited* knowledge.

In our working example, we exemplify an expert recommendation process with five experts in the requirements engineering domain. Table 3.4 shows a recommended list of experts ranked in descending order of the expertise level. Besides, stakeholders who want to delegate evaluations to other experts can select different experts for different requirements. As shown in Table 3.1, the development of the defined requirements requires a deep knowledge of different areas. Therefore, selecting an appropriate expert for each requirement helps to increase the overall quality of requirements engineering. The expert selection for stakeholders S_2, S_4, and S_5 are depicted in Table 3.5. In this table, we can observe that stakeholders select different experts for different requirements. For instance, stakeholder S_2 requires experts' evaluations for the dimensions of requirements R_1, R_2, and R_3. He chooses $Expert_3$ for R_1, $Expert_2$ for R_2, and $Expert_4$ for R_3. Furthermore, we can observe that this stakeholder does not need any expert for R_4 and this is represented by a dash symbol ('-') in Table 3.5.

After the selection of experts for each requirement, these experts evaluate the remaining requirement dimensions which were not evaluated by the stakeholders S_2, S_4, and S_5. The evaluations which were given by experts are shown in bold numbers in Table 3.6. Next, the utility of each requirement has to be calculated and used as one of the crucial criteria to assign the requirements to releases. The utility of each requirement is calculated based on *Multi-attribute Utility Theory* (*MAUT*) (see Section 3.5).

3.5. Application of Multi-attribute Utility Theory

As already mentioned before, configurable items are usually described by a set of dimensions/attributes. In this context, *Multi-attribute Utility Theory* (*MAUT*) (Dyer, 2005) can be applied. In this chapter, we propose a new *MAUT-based approach* that calculates the utility of an item i according to the evaluations of stakeholders (*evaluation(s,d)*) with regard to dimensions d, the importance of these dimensions (*w(s,d)*) from the stakeholders' point of view, and the importance of stakeholders/experts (*w(s)*). The final result of the MAUT evaluation is then represented by the *weighted average* of all stakeholders' evaluations for the dimensions d.

Formula 3.1 indicates that the evaluation of an expert e for a dimension d (*evaluation(e,d)*) is used in cases where a stakeholder's voting is delegated. Otherwise, the stakeholder's evaluation will be taken into

Requirements		Stakeholders				
		S_1	S_2	S_3	S_4	S_5
R_1	*risk*	5	3	3	**2**	3
	profit	3	3	4	**3**	3
	effort	4	**3**	3	**3**	4
R_2	*risk*	2	2	5	4	**3**
	profit	-	**3**	-	**4**	**2**
	effort	4	3	4	2	4
R_3	*risk*	4	2	2	2	2
	profit	4	5	3	4	**4**
	effort	4	**3**	3	4	4
R_4	*risk*	2	2	4	1	4
	profit	-	5	2	3	3
	effort	-	4	-	**5**	4

Table 3.6.: The evaluations of stakeholders for the defined requirements in Table 3.1. Each requirement is represented by the three following properties: *risk, profit,* and *effort*. The evaluation is in the range of 1 to 5 (5 indicates a requirement with *low risk, high profit,* and *low effort* and 1 represents a requirement with *high risk, low profit,* and *high effort*). Evaluations which were not provided by stakeholders or experts are represented by dash symbols ('-'). Evaluations provided by experts are represented in bold numbers.

account for the MAUT calculation. In our approach, compared to a stakeholder, an expert has a higher impact on the overall utility of an item. Particularly, the weight of an expert is *twice* the weight of a stakeholder (see Formula 3.2). In addition, the expertise level *el(e)* of an expert *e* is also considered in the weight calculation. The total MAUT value (i.e., the utility value) of a requirement R_i is then calculated by summing all dimension-specific MAUT values of the requirement R_i (see Formula 3.3).

$$eval(s,d) = \begin{cases} evaluation(e,d) & \text{if } \textit{evaluation(s,d) delegated} \\ evaluation(s,d) & \textit{otherwise} \end{cases} \tag{3.1}$$

$$w(s) = \begin{cases} weight(s) * 2 + el(e) & \text{if } \textit{evaluation(s,d) delegated} \\ weight(s) & \textit{otherwise} \end{cases} \tag{3.2}$$

$$Utility(R_i) = \frac{\sum_{s \in stakeholders} \frac{\sum_{d \in dims} eval(s,d) * w(s,d) * w(s)}{\sum_{d \in dims} w(s,d) * w(s)}}{|stakeholders|} \tag{3.3}$$

An example of the utility calculation of a requirement is presented in Formula (3.4). In this example, for simplicity, we assume that all stakeholders assign the same weight (i.e., the weight of 1) to all dimensions of the requirements ($\forall s \in stakeholders$, $\forall d \in dimensions$, w(s,d) = 1). Additionally, we assume each stakeholder has the same importance ($\forall s \in stakeholders$ weight(s) = 1).

$$
\begin{aligned}
Utility(R_2) &= \frac{\sum_{s \in stakeholders} \frac{\sum_{d \in dims} eval(s,d)*w(s,d)*w(s)}{\sum_{d \in dims} w(s,d)*w(s)}}{|stakeholders|} \\
&= \frac{1}{5}\Big(\frac{2*1+4*1}{1+1} + \frac{2*1+3*(1*2+4.5)+3*1}{1+(1*2+4.5)+1} \\
&+ \frac{5*1+4*1}{1+1} + \frac{4*1+4*(1*2+4.5)+2*1}{1+(1*2+4.5)+1} \\
&+ \frac{3*(1*2+3.75)+2*(1*2+3.75)+4*1}{(1*2+3.75)+(1*2+3.75)+1}\Big) \\
&= \frac{1}{5}\Big(\frac{6}{2} + \frac{24.5}{8.5} + \frac{9}{2} + \frac{32}{8.5} + \frac{32.75}{12.5}\Big) = 3.354
\end{aligned}
\tag{3.4}
$$

Similarly, the MAUT values (i.e., the utility values) of other requirements are also calculated using Formulae 3.1 - 3.3. The MAUT values of requirements R_1, R_2, R_3, and R_4 are the following: $MAUT(R_1) = 3.266$, $MAUT(R_2) = 3.354$, $MAUT(R_3) = 3.380$, and $MAUT(R_4) = 3.326$. After the calculation of requirement utilities, requirements will be assigned to defined releases (see *Section 3.6*).

3.6. Release Planning

Section 3.5 shows how the utility value of a requirement can be calculated based on *Multi-attribute Utility Theory* (*MAUT*). *The higher the MAUT value, the sooner the requirement will be implemented.* In the context of requirements engineering, making a requirement recommendation is referred to as release planning, i.e., to clarify which requirement should be implemented in which release. In release planning, stakeholders have to estimate the invested effort for each requirement. In our working example, the *effort* refers to the invested time (in hours) to implement a requirement. *The higher the evaluation of effort, the lower the invested time.* We assume that an evaluation of 5 corresponds to an effort of 50 hours, whereas an evaluation of 1 corresponds to an effort of 250 hours. To calculate the effort of a requirement, we first calculate the average of all stakeholders' evaluations concerning the requirement's effort. After that, the effort is calculated using the Formula 3.5, where $effort(R_i, s)$ is the evaluation of stakeholder s for the effort of R_i.

$$
effort(R_i) = \Big(5 - \frac{\sum_{s \in stakeholders} effort(R_i, s)}{|stakeholders|} + 1\Big) * 50 \tag{3.5}
$$

We exemplify the effort calculation of requirement R_1 as shown in Formula 3.6. The effort values of other requirements are calculated in a similar way and presented in Table 3.7.

$$
effort(R_1) = \Big(5 - \frac{4+3+3+3+4}{5} + 1\Big) * 50 = 130 \tag{3.6}
$$

In our example, release planning is done based on four criteria: (*1*) the *utility* (MAUT) value, (*2*) the *effort* (in hours), (*3*) the *dependency* between requirements, and (*4*) the *capacity* of releases. Given the fact that requirement R_3 achieves the highest utility (i.e., $MAUT(R_3) = 3.380$) and its estimated time effort of 120 hours, R_3 turns out to be the best candidate to be assigned to *Release 1*. Furthermore, it is reasonable that the requirement R_2 should follow R_3 and hence it also be assigned to *Release 1*. Indeed, R_2 shows the second highest utility and the remaining capacity of *Release 1* is enough to cover R_2 (the capability of Release 1 is 260 hours - see Table 3.2). Next, requirement R_4 has to be assigned to some release. This requirement can not belong to *Release 1* since its effort (83.33 hours) exceeds the remaining time of *Release*

Requirement	Average effort (*effort in hours*)	Assigned release
R_1	3.4 (130)	*Release 2*
R_2	3.4 (130)	*Release 1*
R_3	3.6 (120)	*Release 1*
R_4	4.33 (83.33)	*Release 2*

Table 3.7.: The assignment of the requirements to releases based on the *effort* of requirements, *dependencies* between requirements, their *utility values*, and *the capacity of releases*. The effort of each requirement is represented in the second column of the table.

1 (10 hours). Finally, the requirement R_1 is assigned to the second release. Based on the requirements' description shown in Table 3.1, we can observe that there is a dependency between R_1 and R_2, which is indicated as follows: *"The evaluation software requires the access to the clock's internal memory"*. This means R_1 (i.e, evaluation software) can not be implemented before R_2 (i.e., data storage function). In our example, the identified dependency does not trigger any changes since the release plan in Table 3.7 shows that requirement R_2 which is assigned to the first release (development period: from 2020-05-01 to 2020-07-01) will be implemented before the requirements assigned to the second release (development period: from 2020-07-15 to 2020-09-15). With this final step, all requirements are assigned to the releases, and the requirements engineering process is therefore complete.

3.7. Conclusion and Future Work

In this chapter, we introduced utility analysis concepts that focus on *liquid democracy*. These concepts allow the manual delegation of a stakeholder's voting right to a domain expert. First, we described a scenario for the development of a sport watch which is used as a working example throughout the chapter. Based on the working example, we applied liquid democracy in order to receive consultations from experts in situations where stakeholders do not have enough knowledge with regard to specific requirements. Afterward, we proposed a novel approach of MAUT-based evaluation which takes into account the evaluations of both users and experts and assigns higher importance to expert consultations (i.e., evaluations). Finally, we proposed a group-based configuration for release planning where requirements were assigned to releases based on derived utility values, effort estimations, existing dependencies, and release capacities.

Within the scope of future work, we plan to integrate the proposed approach in a requirements engineering tool named OPENREQ!LIVE*. It is a *modern innovative release planning tool* which makes use of intelligent techniques to facilitate the requirements engineering process. In the current version of OPENREQ!LIVE, stakeholders can evaluate the requirements without the support of domain experts. However, in the future, we will integrate our approach into this tool to improve the quality of requirements engineering.

*https://live.openreq.eu/

Chapter 4

Explanations for Groups

The contents of this chapter were published in (Felfernig et al., 2018d). For this publication, the author of the thesis provided major contributions in terms of literature research and writing.

4.1. Abstract

Explanations are used in recommender systems for various reasons. Users have to be supported in making (high-quality) decisions more quickly. Developers of recommender systems want to convince users to purchase specific items. Users should better understand how the recommender system works and why a specific item has been recommended. Users should also develop a more in-depth understanding of the item domain. Consequently, explanations are designed in order to achieve specific goals, such as increasing the transparency of a recommendation or increasing a user's trust in the recommender system. In this chapter, we provide an overview of existing research related to explanations in recommender systems and specifically discuss aspects relevant to group recommendation scenarios. In this context, we present different ways of explaining and visualizing recommendations determined on the basis of *aggregated predictions* and *aggregated models* strategies.

4.2. Introduction

Explanations have been recognized as an essential means to deliver persuasive messages to users, help them to evaluate recommendations, and make better decisions (Herlocker et al., 2000; Tintarev and Masthoff, 2011). Empirical studies show that users appreciate explanations of recommendations (Herlocker et al., 2000; Cramer et al., 2008). Explanations can be regarded as *a means to make something clear by giving a detailed description* (Tintarev and Masthoff, 2012). In the recommender systems context, Friedrich and Zanker (Friedrich and Zanker, 2011) define explanations as *information about recommendations* and *as means to support objectives defined by the designer of a recommender system*. Explanations can be seen from two fundamental viewpoints (Bilgic and Mooney, 2005; Tintarev et al., 2016): (1) the *user's* (*group member's*) and (2) the *recommender provider's* point of view. *Users of recommender systems* require additional information to be able to develop a better understanding of the recommended items. *Developers of recommender systems* want to provide additional information to users for various reasons, for example, to convince the user to purchase an item, to increase a user's item domain knowledge (educational aspect), and to increase a user's *trust* in and overall satisfaction with the recommender system. Another objective is to make users more *tolerant* of recommendations provided by the system. This is especially important for new users/items. Otherwise, a recommendation may be perceived as inappropriate. Solely providing the core functionality of recommender systems, i.e., showing a list of relevant items to users, could evoke the impression of interacting with a *black box* with no transparency and no additional user-relevant information (Herlocker et al., 2000; Tintarev and Masthoff, 2011). Consequently, explanations are an important means to provide information related to recommendations, the recommendation generation process, and further objectives defined by the designer of a recommender system (Friedrich and Zanker, 2011; Chen and Pu, 2012; Verbert et al., 2013; Lamche et al., 2014; Quijano-Sanchez et al., 2017). Visualizations of explanations can further improve the perceived quality of a recommender system (Gansner et al., 2009; Verbert et al., 2013; Tintarev et al., 2016) - where appropriate, examples of visualizations will be provided.

In this chapter, we provide an overview of existing research concerning explanations in recommender systems and especially focus on discussing relevant aspects in group recommendation scenarios. The remainder of the chapter is organized as follows. In *Section 4.3*, we summarize existing approaches of generating explanations in recommender systems for single users and groups. In *Sections 4.4, 4.5, 4.6*, and *4.7*, we respectively present different ways of explaining and visualizing recommendations in the context of *collaborating filtering, content-based filtering, constraint-based,* and *critiquing-based recommendation*. Finally, in *Section 4.8*, we conclude the chapter and discuss some further issues for future work.

4.3. Explanations in Recommender Systems for Single Users and Groups

4.3.1. Explanations in Single-User Recommender Systems

In single-user recommender systems, various efforts have already been undertaken to categorize explanations with regard to *information sources used to generate explanations* and corresponding *goals of explanations* (Tintarev, 2009; Tintarev and Masthoff, 2011; Friedrich and Zanker, 2011; Gedikli et al., 2014; Tintarev and Masthoff, 2015; Nunes and Jannach, 2017). A categorization of different information sources that can be used for the explanation of recommendations is given, for example, in (Friedrich and Zanker, 2011) where *recommended items, alternative items,* and the *user model* are mentioned as three orthogonal *information categories*. Potential goals of explanations are discussed in (Tintarev and Masthoff, 2011) and (Jameson et al., 2015). Examples thereof are *efficiency* (reducing the time needed to complete a choice task), *persuasiveness* (exploiting explanations to change the user's choice behavior) (Gedikli et al., 2014), *effectiveness* (proactively helping the user make higher-quality decisions), *transparency* (reasons as to why an item has been recommended, i.e., answering why-questions), *trust* (supporting the user in increasing her confidence in the recommender), *scrutability* (providing ways to make the user profile manageable), *satisfaction* (explanations focusing on aspects such as enjoyment and usability), and *credibility* (assessed likelihood that a recommendation is accurate). Bilgic and Mooney (2005) offered a differentiation between explanations that focus on (1) *promotion*, i.e., convincing users to adopt recommendations, and (2) *satisfaction*, i.e., to help users make more accurate decisions.

Examples of verbal explanations for single-user recommendations include phrases such as (1) *"users who purchased item X also purchased item Y"*, (2) *"since you liked the book X, we recommend book Y from the same authors"*, (3) *"since you prefer taking sports photos, we recommend camera Y because it supports 10 pics/sec in full-frame resolution"*, and (4) *"item Y would be a good choice since it is similar to the already presented item X and has the requested higher frame rate (pics/sec)"*. These example explanations are formulated based on information collected and provided by the underlying recommendation approaches, i.e., (1) collaborative filtering, (2) content-based filtering, (3) constraint-based recommendation, and (4) critiquing-based recommendation - see, for example, (Herlocker et al., 2000; Chen and Pu, 2012; Felfernig et al., 2008b; Gkika and Lekakos, 2014). These examples of explanations can be regarded as *"basic"* since further information could be included. For instance, information related to competitor items and previous user purchases: *"Since you prefer taking sports photos, we recommend camera Y because it supports 10 pics/sec in full-frame resolution. Z would have been the other option, but we propose Y since you preferred purchasing from provider k in the past and Y is only a little bit more expensive than its competitors"*.

Another type of explanation is the following: *"No solution could be found - if you increase the maximum acceptable price or decrease the minimum acceptable resolution, a corresponding solution can be identified"*. This explanation focuses on indicating options to find a way out of the *"no solution could be found"* dilemma, which primarily occurs in the context of constraint-based recommendation scenarios (Felfernig and Burke, 2008). Another example is *"item Y outperforms item Z in both quality and price, whereas X outperforms Z only in quality"*. This explanation does not focus on one item but supports the comparison of different candidate items (in this case, *X* and *Y*). Importantly, it is directly related to the concept of *asymmetric dominance* (*Y* outperforms *Z* two times whereas *X* does this only once) which is a *decision bias* discussed in (Felfernig et al., 2018b). Explanations based on item comparisons are mostly supported in critiquing-based (Chen and Pu, 2012) and constraint-based recommendation (Felfernig et al., 2007b) which are both based on semantic recommendation knowledge. In critiquing-based recommendation, *compound critiques* point out the relationship between the current reference item and the corresponding candidate items (McCarthy et al., 2004b). An example of a compound critique in the domain of *digital cameras* is the following: *"On the basis of the current reference item X, you can take a look at cameras with a [lower price] and a [higher resolution] or at cameras with a [higher price] and a [higher optical zoom]"*. An analysis of comparison interfaces in single-user constraint-based recommendation is presented in (Felfernig et al., 2007b, 2014a).

4.3.2. Explanations in Group Recommender Systems

The aforementioned explanation approaches focus on single users and therefore, do not have to consider certain aspects of group decision making. Explanations for groups can have *further goals* such as *fairness* (taking into account, as far as possible, the preferences of all group members), *consensus* (group members agree on the decision), and *optimality* (a group makes an optimal or nearly-optimal decision*). An important aspect in this context is that explanations show how the interests of individual group members are taken into account. This is not relevant in the context of single-user recommender systems. Understanding the underlying process enables group members to evaluate the appropriateness of the way their preferences have to be taken into account by the group recommender system. Similar to explanations for single users, explanations for groups are shaped by the underlying recommendation algorithms. Explanations similar to those already mentioned can also be defined in a group context. For example, (1) *"groups that like item X also like item Y"*, (2) *"since the group likes the film X, we also recommend film Y from the same director"*, (3) *"since the maximum camera price accepted by group members is 500 (defined by Paul) and the minimum accepted resolution is 18 mpix (defined by Joe), we recommend Y which supports 20 mpix at a price of 459"*, and (4) *"item X is a good choice since it supports a higher frame rate requested by all group members and is only a little bit more expensive"*.

These examples show that the chosen preference aggregation approach (Masthoff, 2011; Felfernig et al., 2018a) has an impact on the explanation style. While *aggregated predictions* include information about the individual preferences of group members (e.g., one group member specified the lowest maximum price of 500) and thus support explanation goals such as *fairness* and *consensus*, *aggregated models*-based

*In contrast to single-user decision making, the exchange of decision-relevant knowledge among group members has to be fostered (Atas et al., 2017) approaches restrict explanations to the group level (e.g., *"groups that like X also like Y"*). More advanced (hybrid) explanations (Kouki et al., 2017) can also be formulated in group recommendation scenarios, for example, *"since all group members prefer sports photography, we recommend camera Y rather than camera Z. It is only a little bit more expensive but has higher usability, which is important for group member Joe who is a newbie in digital photography. Similar groups also preferred Y"*.

An example of an explanation in a situation where no solution could be found is: *"No 23 mpix camera with a price below $250 could be found. Therefore, we recommend camera Y with 20 mpix and a price of $249 since the price is the most important criterion for all group members"*. Finally, the following example shows how to take into account a group's social reality, for example, in terms of *"tactful"* explanations (Quijano-Sanchez et al., 2017): *"Although your preference for item Y is not very high, your close friend Peter thinks it is an excellent choice"*. This example explanation is formulated on the level of *aggregated predictions* (Felfernig et al., 2018a) and also takes into account social relationships among group members (e.g., neighborhoods in a social network). On the level of *aggregated models*, an explanation can be formulated as follows: *"A majority thinks that it is a good choice. Some group members think that it is an excellent choice."* (assuming the existence of at least some aggregated categorization of preferences such as the number of likes). Taking into account the individual preferences of group members helps to increase *mutual awareness* among group members, and thus counteracts the natural tendency to focus on one's own favorite alternatives (Jameson and Smyth, 2007). An approach to explaining the *consequences of a given recommendation* was introduced by Jameson et al. (2004), where *emotions* of individual group members with regard to a recommendation are visualized in terms of animated characters.

We want to emphasize that *explanations for groups* are a highly relevant research topic with a limited, but nevertheless direction-giving, number of research results (Ardissono et al., 2003; Jameson, 2004; Jameson and Smyth, 2007; Chen, 2011; Ntoutsi et al., 2012). In the next sections, we sketch ways in which explanations for single-user recommendation scenarios can be adapted to groups. Following the idea of categorizing explanation types along the different recommendation approaches (Vig et al., 2009; Tintarev and Masthoff, 2012), we discuss explanations for groups in the context of *collaborative-* and *content-based filtering*, as well as *constraint-* and *critiquing-based* recommendation.

4.4. Collaborative Filtering

A widely used example of explanations in collaborative filtering recommenders is *"users who purchased item X also purchased item Y"*. Such explanations can be generated, for example, on the basis of *association rule mining* which is often used as a model-based collaborative filtering approach (Lin et al., 2002). Herlocker et al. (2000) analyzed the role of explanations in collaborative filtering recommenders. They focused on the impact of different explanation styles on user acceptance of recommender systems. Explanations were mostly represented graphically. For example, a histogram of neighbors' ratings for the recommended item categorized ratings as *"good"*, *"neutral"*, or *"bad"*. The outcome of their study was that rating histograms are the most compelling way to explain rating data. Furthermore, *simple graphs* were perceived as more compelling than more detailed explanations, i.e., *simplicity of explanations is a key factor*.

An orthogonal approach to proposing explanations for collaborative-filtering-based recommendations was presented by Chang et al. (2016). Following the idea of generating recommendations based on knowledge from the crowd (see, e.g., (Ulz et al., 2017)), the authors introduced the idea of asking crowd workers to provide feedback on explanations. Quality assurance is an issue, but *crowd-sourced explanations* were considered high quality. The authors mentioned that *longer explanation texts* and an *increased number* *of references to item genres* as examples of indicators of high-quality explanations. An example of a question for crowd-sourcing in group recommendation scenarios is the following: *"Given this movie recommendation (e.g., Guardians of the Galaxy), which of the following are useful explanations for a group of middle-aged persons? Can be viewed by the whole family; Includes plenty of songs from the 70ies; Best movie we have ever seen"*. This way, crowd knowledge can be exploited to better figure out which kinds of explanations are useful in which context and which ones might be particularly well-received by specific groups (in this case, middle-aged persons). A similar approach can be used to figure out relevant explanations in other recommendation approaches, i.e., *which tags to use for an explanation?* (content-based filtering), *which requirements to relax?* (constraint-based recommendation), and *which critiques to propose to the user?* (critiquing-based recommendation).

As mentioned by Bilgic and Mooney (2005), a goal of the explanations introduced in (Herlocker et al., 2000) is to promote items but not to provide more insights into why the items have been recommended, i.e., not to provide satisfaction-oriented explanations that might help users make more accurate decisions. There are different ways to move the explanation focus towards more informative explanations. As proposed in (Bilgic and Mooney, 2005) (for single-user recommenders), a collaborative-filtering-based explanation can be extended by providing information on items that had a significant influence on the determination of the proposed recommendation. Removing the *most influential items* (already rated by group members) from the set of rated items triggers the most significant difference in terms of recommended item ratings. Similar approaches can be used to determine the most influencing items in other recommender types (Bilgic and Mooney, 2005; Symeonidis et al., 2008).

Collaborative Filtering Explanations for Groups

An example of basic explanations in group-based collaborative filtering is included in POLYLENS where the predicted ratings for each group member and for the group as a whole are shown (O'Connor et al., 2001). Some simple examples of how to provide explanations in the context of group-based collaborative filtering scenarios are provided in Tables 4.1 and 4.2. Both examples represent variants of the explanation approaches introduced by Herlocker et al. (2000). Table 4.1 depicts an example of an explanation that is based on the preferences (ratings) of the nearest neighbors ($NN = \bigcup\{n_{ij}\}$) of the group members u_i (for simplicity, we assume the availability of a complete set of rating data). For each recommended item t_i, the corresponding frequency distribution of the ratings of the nearest neighbors of individual group members is shown. Note that *NN* can represent users who are in the *intersection* of users who rated this item ($\{n_{11}, n_{12}, ...\} \cap ... \cap \{n_{m1}, n_{mk}, ...\}$). Alternatively, *NN* can represent the users in the *union* of nearest neighbors ($\{n_{11}, n_{12}, ...\} \cup ... \cup \{n_{m1}, n_{mk}, ...\}$). A related explanation can be *"users similar to members of this group rated item T as follows"*.

Table 4.2 depicts an example of an explanation that is based on the preferences of neighborhood groups gp_j of the current group gp. We assume that ratings are only available in an aggregated fashion (ratings of individual users are not available, e.g., for privacy reasons). In this context, the frequency distribution of the ratings of the nearest neighbor groups is shown for each item t_i. An explanation can contain the following text: *"Groups similar to the current group rated item T as follows"*.

In the given examples, explanations refer to ratings but do not take into account aggregation functions that were used (Felfernig et al., 2018a). Ntoutsi et al. (2012) presented an approach to explaining the aggregation functions in aggregated-prediction-based collaborative filtering. For example, the application of *Least Misery* (LMS) triggers explanations of type *"item Y has a group score of 2.9 due to the (lowest) rating determined for user u_a"*. A more *"group-oriented"* explanation is *"item Y is recommended because it avoids misery within the group"*. When using *Most Pleasure* (MPL), the corresponding explanation

rec. item t_i	**ratings of nearest neighbors** $n_{ij} \in NN$						**explanation**		
	u_1		u_2		u_3		bad	neutral	good
	nn_{11}	nn_{12}	nn_{21}	nn_{22}	nn_{31}	nn_{32}	[0-2]	$[> 2 - 3.5]$	$[> 3.5 - 5]$
t_1	4.2	4.9	4.3	3.5	3.2	4.8	0	2	4
t_2	3.5	2.2	2.7	3.2	2.9	3.6	0	5	1
t_3	3.8	3.1	3.7	2.8	3.4	2.6	0	4	2
t_4	4.3	4.9	4.4	4.5	4.0	4.0	0	0	6
t_5	3.7	3.9	3.2	3.5	3.6	2.9	0	3	3

Table 4.1.: Collaborative filtering explanations for *aggregated predictions*. The explanations are based on information about the preferences (ratings) of nearest neighbors (n_{ij}) of individual group members u_i.

rec. item t_i	**ratings of NN groups**				**explanation**		
	gp_1	gp_2	gp_3	gp_4	bad $[0 - 2]$	neutral $[> 2 - 3.5]$	good $[> 3.5 - 5]$
t_1	4.2	4.9	4.3	3.5	0	1	3
t_2	1.2	2.9	3.1	1.8	2	2	0
t_3	3.5	3.8	2.9	3.3	0	3	1
t_4	4.9	4.8	4.1	4.4	0	0	4
t_5	3.7	3.3	2.4	3.9	0	2	2

Table 4.2.: Collaborative filtering explanations for *aggregated models*, i.e., explanations are based on the aggregated preferences of individual group members.

would be *"item Y has a group score of 4.8 due to the (highest) rating determined for user u_b"*. Finally, when using *Average* (AVG), explanations of type *"item Y is the most similar to the ratings of users u_a, u_b, and u_c"* are provided. Similar explanations can be generated for content-, constraint-, and critiquing-based recommendations. Although initial approaches have already been proposed, different ways to explain group recommendations depending on the used aggregation function(s) are an issue for future research.

Visualization of Collaborative Filtering Explanations for Groups

There are different ways to visualize a recommendation determined by collaborative filtering approach (Herlocker et al., 2000). The frequency distributions introduced and evaluated by Herlocker et al. (2000) can also be applied in the context of group recommendation scenarios. An example thereof is given in Figure 4.1, where the explanation information contained in Table 4.1 is represented graphically. Figure 4.2 depicts a similar example where an item-specific evaluation of the nearest (the most similar) groups is shown in terms of a frequency distribution. Alternatively, *spider diagrams* can be applied to visualize the preferences of nearest neighbors. An example is depicted in Figure 4.3. This type of representation is based on the idea of *consensus-based approaches* to visualize the current status of a group decision process (Palomares et al., 2014b; Mahyar et al., 2017).

4.5. Content-based Filtering

The basis for determining recommendations in content-based filtering is the similarity between item descriptions and keywords (categories) stored in a user profile. Since the importance of keywords can

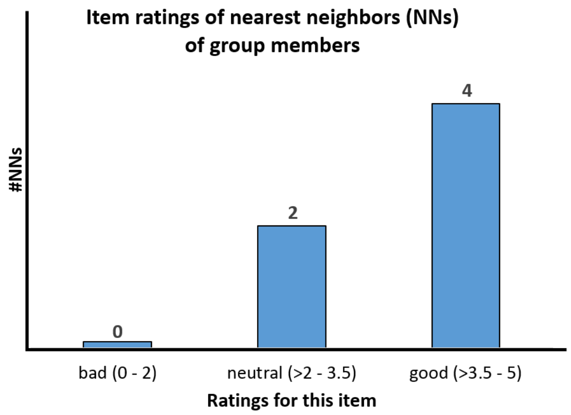


Figure 4.1.: Graphical representation of the explanation data contained in Table 4.1.

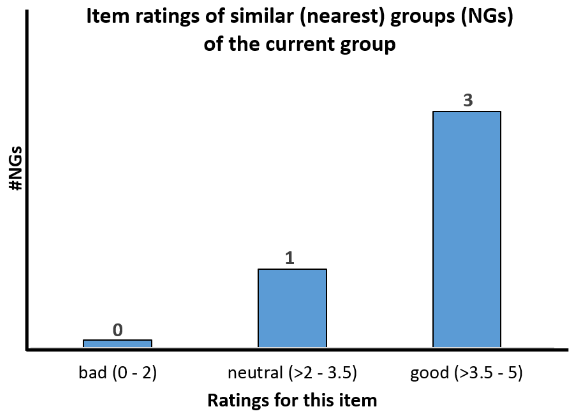


Figure 4.2.: Graphical representation of the explanation data contained in Table 4.2.

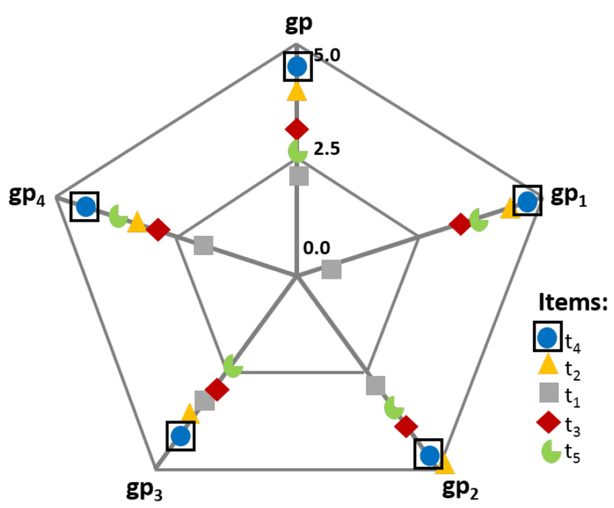


Figure 4.3.: Spider diagram for explaining *aggregated models* based collaborative filtering recommendations: ratings of nearest neighbor groups gp_1,.., gp_4 of gp for the recommended item t_4. This representation is a variant of consensus-based interfaces discussed in (Mahyar et al., 2017).

category	userweights			itemweights				explanation-relevance			
	u_1	u_2	u_3	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4
cat_1	0.05	0.1	0.15	0.1	0.1	0.2	0.3	0.01	0.01	0.02	0.03
cat_2	0.3	0.4	0.5	0.7	0.2	0.2	0.0	0.28$\surd$	0.08$\surd$	0.08	0.0
cat_3	0.15	0.25	0.2	0.1	0.4	0.2	0.3	0.02	0.08$\surd$	0.04	0.06$\surd$
cat_4	0.4	0.3	0.2	0.1	0.2	0.3	0.1	0.03	0.06	0.09$\surd$	0.03

Table 4.3.: Content-based filtering explanations for *aggregated predictions*. The most explanation-relevant categories for an item t_k are marked with $\surd$.

differ among group members, it is important to identify those which are relevant for all group members (Lieberman et al., 1999a). Explanations are based on the analysis of *item-related content*. Examples of verbal explanations in content-based filtering were proposed by (Bilgic and Mooney, 2005). The authors showed that *keyword-style explanations* can increase both the perceived trustworthiness and the transparency of recommendations. Such explanations primarily represent occurrence statistics of keywords in item descriptions (see also (Cramer et al., 2008)). Gedikli et al. (2014) compared different approaches to representing explanations in content-based filtering scenarios, and showed that *tag-cloud-based* graphical representations outperform verbal approaches.

Content-based Filtering Explanations for Groups

A simple example of content-based filtering explanations for groups is depicted in Table 4.3. Item categories cat_j have a user-specific weight (derived, for example, from the category weights of individual user profiles where user u_i is a member of group G). To determine the *explanation relevance* of individual categories, these weights are combined with item-individual weights (see Formula 4.1).

$$explanation\text{-}relevance(cat_j, t_k) = \frac{\sum_{u_i \in G} userweight(u_i, cat_j) \times itemweight(t_k, cat_j)}{|G|} \tag{4.1}$$

The higher the explanation-relevance of a category, the higher the category will be ranked on a list shown to the group (members). A verbal explanation related to item t_1 (Table 4.3) can be of the form *"item t_1 is recommended since each group member is interested in category cat_2"*. If the preference information of individual group members is not available (e.g., for privacy reasons), this explanation would be formulated as *"item t_1 is recommended since the group as a whole is interested in category cat_2"*. Also, more than one category can be used in such an explanation. As mentioned, *category-* or *keyword-based* explanations can also be extended with information about the most influential items (Bilgic and Mooney, 2005). This can be achieved by determining those items that trigger the most significant change in item rating predictions (if not taken into account by the recommendation algorithm).

An approach to explaining recommendations based on tags was presented in (Vig et al., 2009). *Tagsplanations* (explanations based on user community tags) are introduced to explain recommendations. In this context, *tag relevance* is defined as the *Pearson Correlation* (Ricci et al., 2010) between item ratings and corresponding tag preference values. *Tag preference* is the relationship between the number of times a specific tag has been applied to an item compared to the total number of tags applied to the item (weighted with corresponding item ratings). In a study with MovieLens (Miller et al., 2004) users, the authors showed that both tag relevance and tag preference help to achieve the explanation goals of *justification* (why an item has been recommended) and *effectiveness* (better decisions are made). Similar to the example shown in Table 4.3, *explanation-relevance* (in this case tag relevance) is used to order a list of explanatory tags (Vig et al., 2009).

An *opinion mining* approach to generating explanations was introduced by Muhammad et al. (2016). In the context of opinion mining, features are extracted from item reviews (Dong et al., 2013) and then associated with corresponding sentiment scores. Features and corresponding sentiments are then used to generate explanations related to the *pros* and *cons* of specific items. Features are sorted into *pros* or *cons* according to whether their values are above or below a predetermined *threshold*. If we assume, for example, a threshold of 0.4, then all item features with an explanation relevance ≥ 0.4 are regarded as *pros*, the others are regarded as *cons*. Formula 4.2 represents an approach to determining the explanation-relevance of a specific feature f_i where *sentiment* represents a group preference for a specific feature and *item-sentiment* represents the support of the feature by the item t_j.

$$explanation\text{-}relevance(f_i) = sentiment(f_i) \times itemsentiment(t_j, f_i) \tag{4.2}$$

Opinion mining approaches to explanations can also be extended to groups. An example of applying Formula 4.2 in the context of group recommender systems is given in Table 4.4.

group profile (*gp*)		**item-sentiments**				**explanation-relevance**			
feature	*sentiment*	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4
f_1	0.10	0.19	0.23	0.35	0.68	0.02	0.02	0.04	0.07
f_2	0.76	0.61	0.52	0.47	0.52	0.46	0.40	0.36	0.40
f_3	0.21	0.47	0.43	0.21	0.31	0.10	0.09	0.04	0.07
f_4	0.82	0.92	0.76	0.49	0.77	0.75$\surd$	0.62$\surd$	0.40$\surd$	0.63$\surd$

Table 4.4.: Opinion mining based explanations for *aggregated models*. Feature f_i with the highest explanation-relevance are marked with $\surd$.

This example sketches the generation of explanations in *aggregated models* scenarios (Felfernig et al., 2018a). When determining explanations in the context of *aggregated predictions*, explanation relevance could be determined for each user and then aggregated using an aggregation function such as *Average* (*AVG*) to select explanations considered most relevant for the group.

Visualization of Content-based Filtering Explanations for Groups

An alternative to list-based representations of explanations was mentioned, for example, in (Gedikli et al., 2014), where content-based explanations were visualized in the form of *tag-clouds*. An example of a *tag-cloud-based explanation* in the context of group recommendation is depicted in Figure 4.4. The used tags are related to a working example from the travel domain (Felfernig et al., 2018a). In this scenario, the tag cloud represents an explanation based on the *aggregated preferences* of individual group members. For example, *Leo* and *Isa* like city tours. One can imagine other visual encodings in terms of *shapes, textures,* and *highlightings* (Knutov et al., 2009). Tag relevance can be determined on the basis of a tag relevance estimator similar to Formula 4.1.

4.6. Constraint-based Recommendation

Constraint-based recommender systems are built upon deep knowledge about items and their corresponding recommendation rules (constraints). This information serves as a basis for explaining item recommendations by analyzing reasoning steps that led to the derivation of solutions (items) (Friedrich and Zanker, 2011). Such explanations follow the tradition of *AI-based expert systems* (Buchanan and Shortliffe, 1985; Friedrich, 2004). On the one hand, explanations are used to answer *how*-questions, i.e.,

Figure 4.4.: Tag-cloud representation used to show the relevance of tags with regard to a specific item extended with preference information related to group members (Isa, Joe, and Leo).

questions related to the reasons behind a recommendation. A corresponding analysis was provided, for example, by Felfernig et al. (2007b). How questions are answered in terms of showing the relationship between defined user requirements req_i and the recommended items. An example of such an explanation is *"item Y is recommended since you specified the upper price limit with \$500 and you preferred light-weight cameras"* (for details see (Felfernig et al., 2007b; Friedrich, 2004)). Besides answering how questions, constraint-based recommenders help to answer *why* and *why not* questions. Explanations for the first type are used to provide to the user insights into why specific questions have to be answered, whereas explanations for *why not* questions help the user to escape from the *"no solution could be found"* dilemma (Felfernig et al., 2009). Felfernig et al. (2007b) showed that such explanations can help to increase a user's trust in the recommender application. Furthermore, explanations related to *why not* questions can increase the perception of item domain knowledge.

Explanations in Constraint-based Recommendation for Groups

Formula 4.3 represents a simple example of an approach to determining the explanation-relevance of user requirements in constraint-based recommendation scenarios for groups. A related example is depicted in Table 4.5. The assumption is that all group members have already agreed on the set of requirements $\bigcup req_j$, and each group member has also specified his/her preference in terms of an importance value. An explanation that can be provided to a group in such a context is *"requirement req_3 is considered important by the whole group"*.

$$explanation\text{-}relevance(req_j) = \frac{\sum_{u_i \in G} importance(req_j, u_i)}{|G|} \tag{4.3}$$

Requirement	Importance			Explanation-relevance
	u_1	u_2	u_3	
req_1	0.2	0.3	0.4	0.3
req_2	0.5	0.4	0.1	0.33
req_3	0.3	0.3	0.5	0.37$\surd$

Table 4.5.: Explanation relevance of requirements in constraint-based recommendation (*aggregated models*). The most relevant requirement is marked with $\surd$.

The example explanation shown in Table 4.5 does not take into account *causal relationships* between requirements and items (Friedrich, 2004). For example, if a group agrees that the *price* of a camera has to be *below $1,000* and every camera fulfills this criterion, then the price requirement does not filter out items from the itemset. Hence, there is no causal relationship between a recommendation subset of a given itemset and the price requirement.

Combining Constraints and Utilities

A constraint-based recommendation is usually combined with an additional mechanism that supports the ranking of candidate items. An example thereof is *Multi-Attribute Utility Theory* (*MAUT*) (Winterfeldt and Edwards, 1986) that supports the evaluation of items in terms of *a set of interest dimensions* which can be interpreted as generic requirements. For example, in the digital camera domain, *output quality* is an interest dimension that is related to user requirements such as *resolution* and *sensor size*. Group members specify their preferences with regard to the importance of the interest dimensions dim_i. Furthermore, items t_j have different contributions concerning these dimensions (see Table 4.6).

Dimension	**Importance**			**Contribution**			**Explanation-relevance**		
	u_1	u_2	u_3	t_1	t_2	t_3	t_1	t_2	t_3
dim_1	0.1	0.3	0.1	0.3	0.3	0.1	0.05	0.05	0.02
dim_2	0.6	0.5	0.3	0.3	0.5	0.6	0.14	0.23$\surd$	0.28$\surd$
dim_3	0.3	0.2	0.6	0.4	0.2	0.2	0.15$\surd$	0.07	0.07

Table 4.6.: Explanation relevance of interest dimensions in utility-based recommendation (*aggregated predictions*). The most relevant dimension is marked with $\surd$.

Similar to content-based filtering, the *item-specific explanation relevance* of individual interest dimensions can be determined on the basis of Formula 4.4, where *imp* represents the user-specific importance of an interest dimension dim_i and *con* is the contribution of an item to dim_i.

$$explanation\text{-}relevance(dim_i, t_j) = \frac{\sum_{u_k \in G}(imp(u_k, dim_i) \times con(t_j, dim_i))}{|G|} \tag{4.4}$$

Following this approach, (Carenini and Moore, 2006; Felfernig et al., 2008a; Symeonidis et al., 2008; Teze et al., 2015) showed how to apply utility-based approaches to the selection of *evaluative arguments*[†], i.e., arguments with the highest relevance. In this context, arguments take over the role of the previously-mentioned interest dimensions. Such an approach is provided in the INTRIGUE system (Ardissono et al., 2003), where recommended travel destinations are explained to groups, and arguments are chosen depending on their utility for individual group members or subgroups.

An example of an argument (as an elementary component of an explanation) for a car recommended by a constraint-based recommender is *"very energy-efficient"*, where *energy-efficiency* can be regarded as an interest dimension. The contribution of an item to this interest dimension is high if, for example, the fuel consumption of a car is low. If a customer is interested in energy-efficient cars and a car is energy efficient, the corresponding argument will be included in the explanation (see the example in Table 4.6). An example explanation from another domain (e.g., financial services) is the following: *"Financial service t_1 is recommended since all group members strongly prefer low-risk investments"*. Examples of interest dimensions used in this context are *risk, availability,* and *profit*.

[†]In line with Jameson and Smyth (2007), we interpret arguments as elementary parts of explanations

Consensus in Group Decisions

Situations can occur where the preferences of individual group members become inconsistent (Felfernig et al., 2012a, 2016; Mahyar et al., 2017). In the context of group recommendation scenarios, a *consensus* is defined in terms of a *disagreement* between individual group members regarding item evaluations (ratings) (Amer-Yahia et al., 2009). To provide a basis for establishing consensus, such situations have to be explained and visualized (Jameson, 2004; Mahyar et al., 2017). In this context, *diagnosis methods* (see Chapter 2) can help to determine repair actions. These repair actions propose changes to the current set of requirements (preferences) such that a recommendation can be identified. Such repairs can take into account the individual preferences of group members (Felfernig et al., 2016). The potential of aggregation functions (Felfernig et al., 2018a; Masthoff, 2011) to foster consensus in group decision making was discussed in (Salamó et al., 2012). Concepts to take into account consensus in group decision making were also presented in (Amer-Yahia et al., 2009; Castro et al., 2015, 2018). In scenarios such as software requirements engineering (Ninaus et al., 2014), there are often misconceptions regarding the evaluation/selection of a specific requirement. For example, there could be misconceptions regarding the assignment of a requirement to a software release. An explanation in such contexts indicates possible changes in requirements (assignments) that help to restore consistency. In group-based settings, such *repair-related explanations* help group members understand the constraints of other group members and decide in which way their requirements should be adapted.

User-generated Explanations

User-generated explanations are defined by a group member (typically, the creator of a decision task) to explain, for example, why a specific alternative has been selected. The impact of user-generated explanations in constraint-based group recommendation scenarios was analyzed by Stettinger et al. (2015). The creator of a decision task (prioritization decisions in the context of software requirements engineering) had to explain the decision outcome verbally. In groups where such explanations were provided, this contributed to an increased satisfaction with the final decision and an increased perceived degree of group decision support quality (Stettinger et al., 2015). *User-generated explanations* are not limited to constraint-based recommendation. For example, *crowd-sourcing based* approaches are based on the similar idea of collecting explanations directly from users.

Fairness Aspects in Groups

Fair recommendations in group settings can be characterized as *recommendations without favoritism or discrimination towards specific group members.* The perceived importance of fairness, depending on the underlying item domain, has been analyzed in (Felfernig et al., 2017). An outcome of this study is that in *high-involvement item domains* (e.g., decisions regarding new cars, financial services, and apartments), the preferred preference aggregation strategies (Masthoff, 2011; Felfernig et al., 2018a) differ from *low-involvement item* domains such as restaurants and movies. The latter are often the domains of repeated group decisions (e.g., the same group selects a restaurant for dinner every three months). Groups tend to apply strategies such as *Least Misery* (LMS) in high involvement item domains and to prefer *Average Voting* (AVG) in low-involvement item domains. When recommending packages, the task is to recommend a set of items in such a way that individual group members perceive the recommendation as fair (Serbos et al., 2017). One interpretation of fairness stated by Serbos et al. (2017) is that *"there are at least m items included in the package that a group member likes"*.

An approach to taking into account fairness in *repeated group decisions* was presented by Quijano-Sanchez et al. (2013), where rating predictions are adapted to achieve fairness in future recommendation settings.

This adaptation also depends on the personality of a group member. For example, *"a group member with a strong personality who was treated less favorably last time, will be immediately compensated in the upcoming group decision"*. A similar interpretation of fairness was introduced in (Stettinger, 2014) where fairness was also defined in the context of repeated group decisions, i.e., decisions that repeatedly take place within the same or stable groups (groups with a low fluctuation). Fairness in this context was achieved by introducing functions that systematically adapt preference weights, i.e., group members whose preferences were disregarded recently receive higher preference weights in upcoming decisions. For example, in the context of repeated decisions (made by the same group) regarding a restaurant for dinner, the preferences of some group members are more often taken into account than the preferences of others. In such scenarios, the preference weights of individual group members can be adapted (Stettinger, 2014) (see Formulae 4.5 and 4.6).

$$imp'(u_i, dim_j) = imp(u_i, dim_j) \times (1 + (\frac{\sum_{u \in G} fair(u)}{|G|} - fair(u_i))) \tag{4.5}$$

$$fair(u_i) = \frac{\#supportedpreference(u_i)}{\#groupdecisions} \tag{4.6}$$

Formula 4.6 provides a *fairness estimate* per user u_i in terms of the share of the *number of supported preferences* in relation to the *number of defined preferences*. The lower the value, the less the preferences of a user (group member of group G) have been considered, and the lower the corresponding degree of fairness with regard to u_i. Formula 4.5 reflects an approach to increasing fairness in *upcoming* recommendation sessions. If the fairness (Formula 4.6) in previous sessions was lower than average, then a corresponding upgrade of user-specific importance weights takes place for each dimension. For an example of adapted weights, see Table 4.7.

User	**Importance (*imp*)**			**Fairness (*fair*)**	**Adapted importance (*imp'*)**		
	dim_1	dim_2	dim_3		dim_1	dim_2	dim_3
u_1	0.3	0.3	0.4	4/8=0.5	0.375	0.375	0.5
u_2	0.5	0.4	0.1	6/8=0.75	0.5	0.4	0.1
u_3	0.3	0.2	0.5	8/8=1.0	0.225	0.15	0.375

Table 4.7.: An example of an adaptation of individual users' weights to take fairness into account. In this example, the importance (*imp*) (or the weights) of user u_1 have been increased, the weights of u_2 remain the same, and the weights of user u_3 have been decreased (the preferences of u_3 have been favored in previous decisions - a visualization is given in Figure 4.6).

Visualization of Constraint-based Explanations for Groups

An example of visualizing the importance of interest dimensions concerning a final evaluation (*utility*) is given in Figure 4.5. Examples of interest dimensions when evaluating, for example, financial services, are *risk, profit,* and *availability*.

If the degree of fairness of previous group decisions has to be made transparent to the group, for example, for explaining adaptations regarding the importance weights of individual group members, this can be achieved on the basis of visualization as depicted in Figure 4.6. An example of a related verbal explanation is the following: *"The interest dimensions favored by user u_1 have been given more consideration in the upcoming decision since she was at a disadvantage in previous decisions"*.

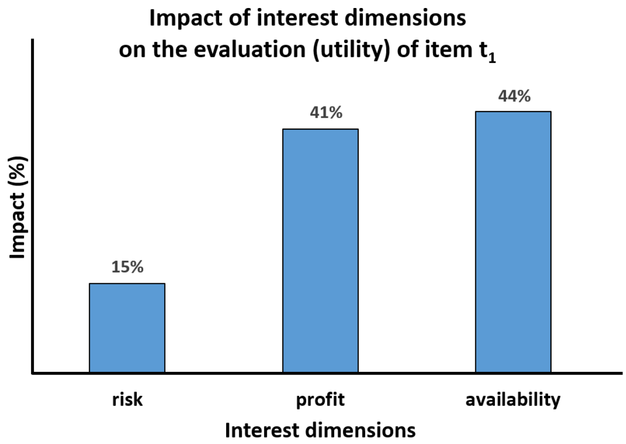


Figure 4.5.: Visualization of the importance of interest dimensions with regard to the overall item evaluation (the importance values are based on Table 4.6 where dim_1 = *risk*, dim_2 = *profit*, and dim_3 = *availability*).

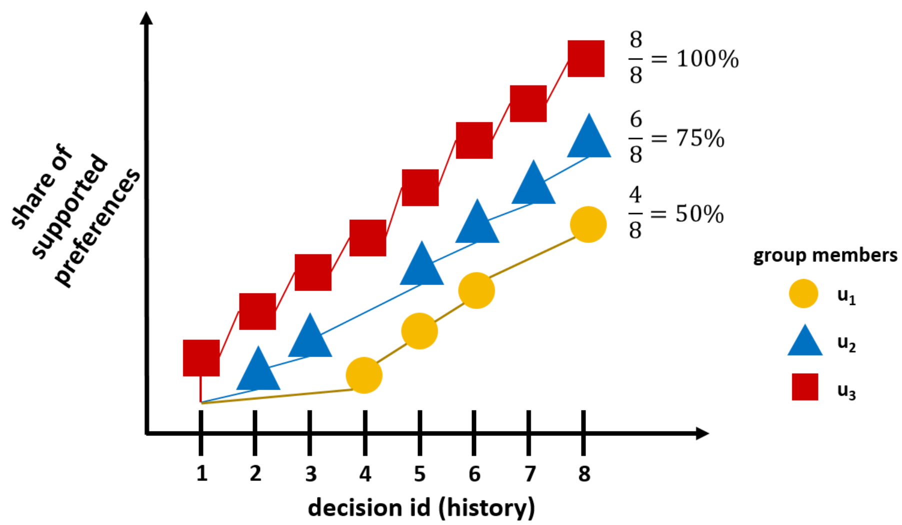


Figure 4.6.: Visualizing the degree of fairness (Formula 4.6) in repeated group decisions (e.g., decisions on restaurant visits). In this example, the visualization indicates that user u_1 was disadvantaged in previous decisions.

4.7. Critiquing-based Recommendation

To assist users in constructing and refining preferences, critiquing-based recommender systems (Chen and Pu, 2012) determine recommendations based on the similarity between candidate and reference items. For example, in the domain of digital cameras, related explanations focus on item attributes such as *price, resolution,* and *optical zoom*. *System-generated critiques* (e.g., compound critiques (McCarthy et al., 2004a)) help to explain the relationship between the currently shown reference item and candidate items. Such explanations have been found which help to educate users and increase their trust in the underlying recommender system (Pu and Chen, 2007).

Critiquing-based Explanations for Groups

User-defined critiques, i.e., critiques on the current reference item directly defined by the user, can be used for the creation of explanations for recommended items (see the example in Table 4.8).

Critiques of group members				$support(attribute, t_i)$		
Attribute	$crit(u_1)$	$crit(u_2)$	$crit(u_3)$	t_1	t_2	t_3
price	$\leq 1,000$	≤ 750	≤ 600	299 (1.0)	650 (0.66)	1,200 (0.0)
resolution	≥ 20	≥ 18	≥ 25	24 (0.66)	25 (1.0)	30 (1.0)
weight	≤ 1	≤ 2	≤ 1	1.5 (0.33)	3 (0.0)	2 (0.33)
exchangeable lens	y	y	n	y (0.66)	y (0.66)	n (0.33)

Table 4.8.: Critiques of group members as a basis for generating explanations for item recommendations. *Support* is defined by the share of *attribute-specific critiques* supported by an item t_i.

In this context, $support(attribute, t_i)$ (see Formula 4.7) indicates how often an item supports a user critique on the attribute. For example, item t_1 supports a critique on *price* three times since all the critiques on price are consistent with the price of t_1, i.e., $support(price, t_1) = 1.0$. However, $support(weight, t_1)$ is only 0.33 since the weight of t_1 is 1.5 which is inconsistent with two related critiques.

$$support(attribute, t_i) = \frac{\#supportedcritiques(attribute, t_i)}{\#critiques(attribute)} \tag{4.7}$$

On the verbal level, an explanation for item t_1 could be: *"The price of camera t_1 (\$299) is clearly within limits specified by the group members. As expected, it has an exchangeable lens. It has a resolution (24 mpix) that satisfies the requirements of u_1 and u_2; however, u_3 has to accept minor drawbacks. Furthermore, the weight of the camera (1.5 kg) is significantly higher than the expectation of u_1 and u_3"*.

Such explanations can be provided if the preferences of group members are known. Otherwise, explanations have to be created on the basis of *aggregated models*, where item properties are compared with the aggregated critiques defined in the group profile.

Visualization of Critiquing-based Explanations for Groups

An example of visualizing the support of different *attribute-specific critiques* is given in Table 4.9. The $\surd$ symbol denotes the fact that the user's critique on an attribute of item t_i is supported by t_i.

User	Attributes(t_1)			
	price = 299	*resolution* = 24	*weight* = 1.5	*exchangeable-lens* = *y*
u_1	$\surd$	$\surd$	$\times$	$\surd$
u_2	$\surd$	$\surd$	$\surd$	$\surd$
u_3	$\surd$	$\times$	$\times$	$\times$

Table 4.9.: Summary of the support-degree of user-specific critiques on item t_1.

4.8. Conclusion and Research Issues

In this chapter, we provided an overview of explanations that help single users and groups to better understand item recommendations. As has been pointed out in pioneering work by Jameson and Smyth (2007), explanations play a crucial role in group recommendation scenarios. We discussed possibilities of explaining recommendations in the context of the basic recommendation paradigms of *collaborative filtering*, *content-based filtering*, *constraint-based*, and *critiquing-based recommendation*, taking into account specific aspects of group recommendation scenarios. In order to support a more in-depth understanding of how explanations can be determined, we provided a couple of working examples of verbal explanations and corresponding visualizations.

Although extensively analyzed in the context of single-user recommendations (see, e.g., Tintarev (2009)), the generation of explanations for groups entails a couple of open research issues. Specifically, aspects of *group dynamics* have to be analyzed concerning their role in generating explanations. For example, *consensus, fairness,* and *privacy* are major aspects - the related research question is *how to define explanations that best help to achieve these goals*. Some initial approaches exist to explain the application of aggregation functions in group recommendation contexts (see, e.g., Ntoutsi et al. (2012)). However, a more in-depth integration of social choice theories into the generation of explanations has to be performed. This is also true on the algorithmic level, as in the context of group-based configuration. In this context, the integration of information about *personality* and *emotion* into explanations has to be analyzed. Initial related work can be found, for example, in (Quijano-Sanchez et al., 2017) where *social factors* in groups are taken into account to generate *tactful explanations*, i.e., explanations that avoid, for example, damaging friendships.

Mechanisms that help to increase the quality of group decision making processes have to be investigated (Konstan and Riedl, 2012). For example, explanations could also be used to trigger intended behaviors in group decision making, such as the exchange of decision-relevant information among group members (Atas et al., 2017). Finally, explaining *hybrid recommendations* (Kouki et al., 2017) and recommendations generated by *matrix factorization* (MF) approaches (Abdollahi and Nasraoui, 2017; Rastegarpanah et al., 2017) are issues for future research. Summarizing, explanations for groups is a highly relevant research area with many open issues for future work.

Chapter 5

Towards Social Choice-based Explanations in Group Recommender Systems

This chapter is based on the results documented in (Tran et al., 2019a). The author of the thesis provided major contributions in terms of literature research, user study, data analysis, and writing all sections of this chapter.

5.1. Abstract

Explanations help users to better understand why a set of items has been recommended. Compared to single-user recommender systems, explanations in group recommender systems have further goals. Examples thereof are *fairness* which helps to take into account as much as possible group members' preferences and *consensus* which persuades group members to agree on a decision. This chapter proposes different explanation types and investigates explanations which best help to increase the *fairness perception, consensus perception*, and *satisfaction* of group members with regard to group recommendations. We conducted a user study to evaluate the proposed explanations. The results show that explanations that take into account preferences of *all* or the *majority* of group members achieve the best results in terms of the mentioned aspects. Moreover, there exist positive correlations among these aspects, i.e., as the perceived fairness (or the perceived consensus) of explanations increases, so does the satisfaction of users with regard to group recommendations. Besides, in the context of repeated decisions, the inclusion of group members' satisfaction from previous decisions in the explanations helps to improve the fairness perception of users concerning group recommendations.

5.2. Introduction

Explanations can be regarded as a piece of information that is presented in a communication process to serve different goals, such as exposing the reasoning behind a recommendation or enabling more advanced communication patterns between sellers and buyers (Jannach et al., 2010). Explanations have been included in recommender systems to help users have an insight into recommendation processes, choose better solutions, and increase the acceptance of recommended items (Tintarev and Masthoff, 2011, 2012; Chen et al., 2013; Felfernig et al., 2018d). Tintarev and Masthoff (2011) discussed potential goals to evaluate explanations in single-user recommender systems. Examples thereof are *transparency* which reveals the underlying mechanism of how a recommendation can be generated, *scrutability* which enables users to check the correctness of recommendations, *effectiveness* which helps users make better decisions, *efficiency* which assists users to make a decision faster, *persuasiveness* which helps users change the choice behavior, and *satisfaction* which increases the acceptance of group members with recommendations.

A majority of studies focus on proposing explanation approaches for single users (e.g., (Herlocker et al., 2000; Chen and Pu, 2012; Gedikli et al., 2014)), but do not take into account certain aspects of group decision making. Although there exist many studies on group recommendations, only a few of them focus on generating explanations in group recommendation contexts. Differing from single-user recommender systems, group recommender systems should take into account not only the preferences of individual users but also the combination of all group members' preferences. This triggers some issues regarding *consensus achieving* (Chiclana et al., 2007), *conflict resolution* (Felfernig et al., 2016; Tran et al., 2016), and *fairness fostering* among group members (Kacprzyk and Zadrożny, 2016). As a result, compared to explanations for single-users, explanations for groups have further goals, such as *fairness* which helps to take into account as much as possible preferences of group members and *consensus* which helps group members agree on the decision (Felfernig et al., 2018d). The explanations taking into account these two aspects could help to increase users' satisfaction with regard to group recommendations.

Fairness and consensus aspects of explanations can be differently considered depending on the *decision type* (e.g., *repeated* or *non-repeated decisions*) (Felfernig et al., 2017). In the context of *non-repeated decisions* in which decisions are rarely repeated (e.g., *"selecting a new house to buy for the whole family"*), these aspects should be taken into account right in the on-going decision. In contrast, for *repeated decisions* which are periodically repeated by the same group (e.g., *"deciding on a restaurant to have dinner together every month"*), these aspects could be considered not only in the on-going decision but also in previous or future decisions. For instance, to foster *fairness* within the group, group members whose preferences have not been considered in previous decisions could have higher priorities in future decisions (Stettinger, 2014). The *consensus* aspect can be defined as an acceptable solution, even if it is not the favorite of every group member (Williams and Mcleod, 2008). An explanation in such a context indicates a solution that makes the final decision more likely to be accepted by every group member (Hertzberg et al., 2013).

In this context, it is questioned that *"how to formulate explanations in such a way that helps to increase the fairness and consensus perceptions of group members concerning group recommendations"*. So far, to some extent, it is still unclear which explanation generation approach could help to improve the mentioned aspects. To address this gap, our focus in this chapter is to propose *social choice-based explanations* and investigate which explanation best helps to increase the *fairness perception, consensus perception,* and *satisfaction* of group members with regard to group recommendations.

The contribution of this chapter is fourfold:

1. We propose *different types of social choice-based explanations* by intuitively explaining the underlying mechanisms of preference aggregation strategies and taking group members' satisfaction into account.
2. We investigate *the best social choice-based explanation* in terms of increasing the fairness perception, consensus perception, and satisfaction of users with regard to group recommendations.
3. We discover *positive correlations* between the perceived fairness/perceived consensus of the explanations and the satisfaction of users with regard to group recommendations.
4. In the context of repeated decisions, we find out that the explanations which take into account group

members' satisfaction from previous decisions increase the fairness perception of users concerning group recommendations.

The remainder of the chapter is organized as follows. In *Section 5.3*, we summarize the related work regarding explanations in group recommender systems. In *Section 5.4*, we introduce different approaches to generate social choice-based explanations. In *Section 5.5*, we define hypotheses and present the main steps of our user study. The results as well as discussions regarding the proposed hypotheses are presented in *Section 5.6*. Finally, we conclude the chapter and discuss open issues for future work in *Section 5.7*.

5.3. Related work

Explanations for groups are usually generated by explaining the underlying recommendation mechanisms (Felfernig et al., 2018d). Ardissono et al. (2003) explained users a recommended tourism attraction by mentioning its positive aspects. For instance, *"the attraction X has been recommended to the group since it is very eye-catching and requires low background knowledge"*. Felfernig et al. (2018d) presented different approaches to explaining group recommendations in the context of collaborative filtering, content-based filtering, constraint-based, and critiquing-based recommendation. Some example explanations are formulated as follows: *"groups that like item X also like item Y"* or *"since the group likes movie X, we also recommend movie Y from the same director"* (Felfernig et al., 2018d). Another approach to generating explanations for groups is to reveal *social choice-based preference aggregation strategies* that are usually applied to generate group recommendations (Senot et al., 2010; Masthoff, 2011; Felfernig et al., 2018a). These strategies allow merging preferences of individual users into a group model that represents the inferred preferences of the whole group. For instance, a textual explanation based on the *Majority* aggregation strategy can be shaped as follows: *"Item X is recommended to the group since most group members like it"*. Regarding this, although Ntoutsi et al. (2012) presented initial approaches to explain aggregation strategies in group recommendation contexts, the integration of social choice theories into the explanation generation is still an open issue.

In addition, as mentioned in *Section 5.2*, group explanations should take into account some social aspects, such as the *fairness* and *consensus* perception of users. In this context, an important concern is that explanations should show how the preferences of group members are considered (Felfernig et al., 2018d). In the current literature, there exist only a few research contributions with an in-depth analysis of this issue. For instance, Kapcak et al. (2018) proposed an approach to generating group explanations in the tourism domain by explaining underlying preference aggregation strategies. After that, an *automated crowd-sourcing pipeline*, which utilizes the *wisdom of crowds*, is used to improve the quality of the generated explanations and increase the satisfaction of users with group recommendations. In another paper, Najafian and Tintarev (2018) proposed explanations based on aggregation strategies and investigated how these explanations can improve users' *satisfaction* with regard to recommended items. The explanations proposed in both mentioned studies can be used only in the context of sequential recommendations. Besides, they have not clearly shown how the proposed explanations could help to increase the fairness and consensus perception of users. To the best of our knowledge, up to now, *'which explanation generation approach performs the best in terms of fairness and consensus aspects'* is still an open issue. To fill this gap, in this chapter, we propose *social choice-based explanations* which not only reveal the underlying mechanisms of preference aggregation strategies but also take into account the fairness perception, consensus perception, and satisfaction of users concerning group recommendations.

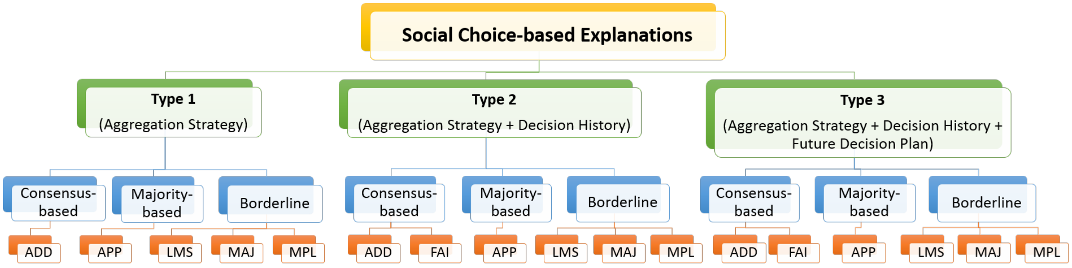


Figure 5.1.: A taxonomy of social choice-based explanations.

5.4. Social Choice-based Explanations

Before proposing social-choice based explanations, we present some preference aggregation strategies which are usually applied to construct a group profile from individual group members' preferences and recommend items to groups.

5.4.1. Social Choice-based Preference Aggregation Strategies

In group recommender systems, *social choice-based preference aggregation strategies* are usually applied to construct a group profile from individual group members' preferences and recommend items to the group. These strategies are grouped into three categories: *consensus-based, majority-based,* and *borderline* strategies (Senot et al., 2010).*

- *Consensus-based strategies* represent preference aggregation mechanisms which take into account preferences of *all individual group members* (Senot et al., 2010; Masthoff, 2011; Felfernig et al., 2018a). We chose *Additive Utilitarian (ADD)* and *Fairness (FAI)* as the representatives of these strategies. The *ADD* strategy recommends an item with the highest total of individual group members' evaluations. The *FAI* strategy is applied in the context of repeated decisions in which the same group of users periodically repeats a decision. This strategy ranks items as if individual group members are choosing them in turn.

- *Majority-based strategies* represent preference aggregation mechanisms that use the *most popular items* among group members (Senot et al., 2010; Felfernig et al., 2018a). We chose *Approval Voting (APP)* as the representative of majority-based aggregation strategies. This strategy recommends an item with the highest number of evaluations which are greater than a *threshold*. The threshold can be pre-defined by the system or by the groups.

- *Borderline strategies* represent preference aggregation mechanisms that take into account only *a subset of individual group members' preferences* (Senot et al., 2010; Felfernig et al., 2018a). We chose *Least Misery (LMS), Majority (MAJ)*, and *Most Pleasure (MPL)* as the representatives of borderline strategies. The *LMS* strategy recommends an item with the highest of all lowest individual evaluations. The *MAJ* strategy recommends an item with the highest of all evaluations representing the majority of item-specific evaluations. The *MPL* strategy recommends an item with the highest of all individual evaluations.

*For a more detailed discussion of the social choice-based preference aggregation strategies, we refer to (Felfernig et al., 2018a).

Explanations		**Templates**
consensus-based	*ADD*	*“Item X has been recommended to the group since it achieves the highest total rating.”*
majority-based	*APP*	*“Item X has been recommended to the group since it achieves the highest number of ratings which are above a threshold* θ*.”*
borderline	*LMS*	*“Item X has been recommended to the group since no group member has a real problem with it.”*
	MAJ	*“Item X has been recommended to the group since most group members like it.”*
	MPL	*“Item X has been recommended to the group since it achieves the highest of all individual group members’ ratings.”*

Table 5.1.: *Type 1* - Explanations based on social choice-based preference aggregation strategies.

5.4.2. Social Choice-based Explanations

We propose *three types* of textual explanations (*Type 1, Type 2,* and *Type 3*). A taxonomy of the proposed explanations is depicted in Figure 5.1.

Type 1 - Based on Preference Aggregation Strategies: Explanations of *Type 1* are generated to explain the underlying mechanisms of social choice-based preference aggregation strategies. Our purpose is to generate *intuitive textual explanations* which help users easily understand how a group recommendation has been generated. We chose *five* aggregation strategies (i.e., *ADD, APP, LMS, MAJ,* and *MPL*) which can be easily represented via verbal/textual explanations. The *FAI* strategy is excluded from *Type 1* since it is usually explained by integrating the information regarding how much the preferences of group members were (or will be) taken into account in previous (or future) decisions. The proposed explanation templates of *Type 1* are presented in Table 5.1.

Type 2 - Based on Preference Aggregation Strategies & Decision History: In the context of repeated decisions, explanations of *Type 2* are formulated by additionally taking into account *decision history* which indicates group members who were treated less favorably in the past decisions. All the selected aggregation strategies (mentioned in *Section 5.4.1*) are applied to formulate the explanations of *Type 2*. The proposed explanation templates of *Type 2* are presented in Table 5.2.

Type 3 - Based on Preference Aggregation Strategies & Future Decision Plan: Similar to *Type 2*, explanations of *Type 3* are also applied to repeated decisions. Each explanation additionally includes a *future decision plan* in which the preferences of disadvantaged group members from the on-going decision will have higher priorities in upcoming decisions. We choose the strategies used for *Type 2* to generate the explanations of *Type 3*. The proposed explanation templates of *Type 3* are presented in Table 5.3.

Explanations		**Templates**
consensus-based	*ADD*	*“Item X has been recommended to the group since it achieves the highest total rating.* ***This decision supports the preferences of users u_a, u_b, and u_c who were treated less favorably in the last n decisions****.”*
	FAI	*“****The preference of user u_a was not considered in the last n decisions. Therefore, in this decision, item X has been recommended to the group since he/she likes it the most****.”*
majority-based	*APP*	*“Item X has been recommended to the group since it achieves the highest number of ratings which are above a threshold* θ*.* ***This decision supports the preferences of users u_a, u_b, and u_c who were treated less favorably in the last n decisions****.”*
borderline	*LMS*	*“Item X has been recommended to the group since no group member has a real problem with it.* ***This decision supports the preferences of users u_a and u_b who were treated less favorably in the last n decisions****.”*
	MAJ	*“Item X has been recommended to the group since most group members like it.* ***This decision supports the preferences of users u_a, u_b, and u_c who were treated less favorably in the last n decisions****.”*
	MPL	*“Item X has been recommended to the group since it achieves the highest of all individual group members’ ratings.* ***This decision supports the preference of user u_a who was treated less favorably in the last n decisions****.”*

Table 5.2.: *Type 2* - Explanations based on social choice-based preference aggregation strategies and decision history.

Explanations		**Templates**
consensus-based	*ADD*	*"Item X has been recommended to the group since it achieves the highest total rating.* ***The preference of user*** u_a ***seems not to be considered in this decision. Therefore, all group members agreed that he/she will have a higher priority in the next decision****."*
	FAI	*"****Item X has been recommended to the group since user*** u_a ***likes it the most. However, all group members agreed that the preferences of other group members will be taken into account in turn in the next decisions****."*
majority-based	*APP*	*"Item X has been recommended to the group since it receives the highest number of ratings which are above a threshold* θ. ***The preference of user*** u_a ***seems not to be considered in this decision. Therefore, all group members agreed that he/she will have a higher priority in the next decision****."*
borderline	*LMS*	*"Item X has been recommended to the group since no group member has a real problem with it.* ***The preferences of users*** u_a ***and*** u_b ***seem not to be considered in this decision. Therefore, all group members agreed that these two users will have higher priorities in the next decisions****."*
	MAJ	*"Item X has been recommended to the group since most group members like it.* ***The preference of user*** u_a ***seems not to be considered in this decision. Therefore, all group members agreed that he/she will have a higher priority in the next decision****."*
	MPL	*"Item X has been recommended to the group since it achieves the highest of all individual group members' ratings.* ***The preferences of users*** u_a, u_b, ***and*** u_c ***seem not to be considered in this decision. Therefore, all group members agreed that these three users will have higher priorities in the next decisions****."*

Table 5.3.: *Type 3* - Explanations based on social choice-based preference aggregation strategies and future decision plan.

5.5. Hypotheses and User Study

In this section, we define hypotheses and present the main steps of our user study.

5.5.1. Hypotheses

One of our goals is to discover the explanation which best helps to increase the *fairness perception, consensus perception*, and *satisfaction* of users with regard to group recommendations. We assume that *ADD-based* explanations would perform the best since compared to other explanations, these explanations describe a group recommendation strategy considering the preferences of *all* group members. In this context, one hypothesis is defined as follows:

Hypothesis H_1: *"ADD-based explanations, which describe a group recommendation strategy taking into account the preferences of all group members, best help to increase the fairness perception, consensus perception, and satisfaction of users with regard to group recommendations"*. The analysis of this hypothesis helps to figure out which explanation approach should be applied to group recommender systems to boost the quality of group recommendations.

Another focus of our study is to answer the following questions: *"Does the perceived fairness (or the perceived consensus) of explanations relate to the satisfaction of users with regard to group recommendations?"* or *"Are they independent of each other?"*. In this context, we investigate whether there exist *positive correlations* between the perceived fairness (or the perceived consensus) of the explanations and the satisfaction of users with group recommendations. One hypothesis is defined as follows:

Hypothesis H_2: *"There exist positive correlations between the perceived fairness (or the perceived consensus) of explanations and the satisfaction of users with regard to group recommendations"*. For this hypothesis, we separately test two relationships: (1) *between perceived fairness and satisfaction* and (2) *between perceived consensus and satisfaction*. The confirmation of this hypothesis helps to prove the fact that *"the higher the perceived fairness (or the perceived consensus) levels of explanations, the higher the satisfaction levels of users with group recommendations"*.

Finally, in the context of repeated decisions, we also try to investigate the influence of the information of *previous decisions* and *future decision plans* on the fairness and consensus perceptions of users about group recommendations. The information of *previous decisions* indicates group members who were treated less favorably from previous decisions. *Future decision plans* indicate how the preferences of less-favored group members from the on-going decision will be taken into account in upcoming decisions. Another hypothesis is defined as follows:

Hypothesis H_3: *"In the context of repeated decisions, the integration of the information regarding group members' satisfaction from previous decisions and future decision plans into social choice-based explanations is assumed to increase the fairness and consensus perception of users with regard to group recommendations"*.

5.5.2. User Study Design

We conducted a user study with staff members and students at two universities[†]. In total, there were *135* user study participants (*male*: 54.81%, *female*: 45.19%) from 20 to 51 years old. The participants were chosen using a *random sampling* method in which each participant had an equal and independent

[†]Graz University of Technology - Austria and Hue University of Economics - Vietnam

	Alex	Anna	Sam	Leo
Rest A	2	2	4	4
Rest B	1	4	4	4
Rest C	5	1	1	1

Table 5.4.: Ratings of group members for the restaurants (1: *the worst*, 5: *the best*).

opportunity of being chosen. Our user study was designed and conducted in the following steps:

Step 1 - Define a group decision scenario: A group decision scenario in the *restaurant* domain was defined as follows:

"Assume, there is a group of four friends (Alex, Anna, Sam, and Leo). Every month, a group decision is made by these friends to decide on a restaurant to have dinner together. To select a restaurant for dinner next month, the group again has to take the same decision. In this decision, each group member explicitly rated three restaurants (*Rest A, Rest B,* and *Rest C*) using a *5-star* rating scale. The ratings given by group members are shown in Table 5.4."

Step 2 - Explanation generation: The explanation templates proposed in *Section 5.4.2* were used to formulate the explanations. Some information in the explanation templates, such as *names of items*, *names of group members*, and *the number of previous decisions*, was accordingly adapted to make them appropriate for the mentioned scenario. For instance, an *ADD-based* explanation of *Type 2* was formulated as follows: *"**Rest B** has been recommended to the group since it achieves the highest total rating. This decision supports the preferences of **Anna, Sam,** and **Leo** who were treated less favorably in the last **three** decisions"*. In total, we formulated *17* explanations (*Type 1*: *five* explanations, *Type 2*: *six* explanations, and *Type 3*: *six* explanations).

Step 3 - Distribute explanations to the participants: We provided each participant with a sequence of *six* different explanations corresponding to six strategies (*ADD, FAI, APP, LMS, MAJ,* and *MPL*). Each explanation in the sequence could be either from *Type 1, Type 2*, or *Type 3*. For instance, for the ADD-based explanation, if this explanation from *Type 1* was already distributed to a user, then this explanation in *Type 2* or *Type 3* will be distributed to the next user. This way, each participant received a different sequence of explanations. Besides, it also made the numbers of participants for each explanation in each type balanced. To avoid possible biases, the explanations in each sequence were shown to the participants in random order. Moreover, at any given time, the participants read and evaluated *"only one"* explanation. The participants' evaluations for the explanations were *independent* of each other, which means the evaluations for one explanation did not reply on those for other explanations.

Step 4 - Evaluate the explanations: Each explanation was evaluated according to three dimensions: *perceived fairness*, *perceived consensus*, and *satisfaction*. For this, each participant received the three following claims:

- **Perceived fairness**: *"The explanation convinces you that the group recommendation is fair to group members"*.
- **Perceived consensus**: *"The explanation helps group members agree on the group recommendation."*
- **Satisfaction**: *"The explanation helps to increase the satisfaction of group members with regard to the group recommendation."*

Each participant provided his/her feedback for the above claims using a *5-point Likert scale* ranging from 1 (*completely disagree*) to 5 (*completely agree*). We want to emphasize that the participant was not a group member of the mentioned group decision scenario. Instead, he/she played the role of a consultant who analyzed the group decision scenario and evaluated the explanations.

5.6. Data Analysis Results and Discussions

In this section, we present data analysis results[‡] and discussions regarding the proposed hypotheses.

5.6.1. Hypothesis H_1

(H_1 - *"ADD-based explanations, which describe a group recommendation strategy taking into account preferences of all group members, best help to increase the fairness perception, consensus perception, and satisfaction of users with regard to group recommendations."*)

Data analysis method: To test the hypothesis H_1, we collected and analyzed the participants' evaluations for the explanations according to the following steps:

- **Step 1**: For each explanation type, we collected *three evaluation sets* corresponding to the *three mentioned dimensions* (i.e., perceived fairness, perceived consensus, and satisfaction) for all explanations. These evaluations share the same characteristics: *ordinal variables* (in the range of [1..5]), *independent* of each other (since the evaluations of one explanation did not rely upon those of other explanations), and *not normally distributed* (*Shapiro-Wilk tests*, significance level $\alpha = .05$, *p values* $< \alpha$).

- **Step 2**: For each explanation type, we performed *three Kruskal-Wallis tests* ($\alpha = .05$) to examine whether there were statistically significant differences in the *perceived fairness, perceived consensus,* and *satisfaction levels* across different explanations.

- **Step 3**: We inspected the *mean ranks* acquired from the Kruskal-Wallis tests to identify the best explanation(s). *The higher the mean rank, the better the explanation in terms of the mentioned dimensions*. In case the mean ranks of some explanations are quite close to each other, we additionally performed follow-up *Mann-Whitney U tests* ($\alpha = .05$) to check whether the evaluations of two certain explanations have the same distributions. For the *five* explanations of *Type 1* (*ADD, APP, LMS, MAJ,* and *MPL*), we performed *10* Mann-Whitney U tests on each dimension. In *Type 2* and *Type 3*, on each dimension, we ran *15* Mann-Whitney U tests between pairs of *six* explanations (*ADD, FAI, APP, LMS, MAJ,* and *MPL*). Running many Mann-Whitney U tests in the same evaluation sets could cause *Type I errors*[§]. To control these errors, we applied a *Bonferroni adjustment* (Pallant, 2007) to adapt the significance level of the Mann-Whitney tests. The revised significance levels were $\alpha' = \alpha/10 = .005$ (for *Type 1*) and $\alpha' = \alpha/15 = .003$ (for *Type 2* and *Type 3*).

Results: The Kruskal-Wallis tests obviously show that in *Type 2*, there were no statistically significant differences in the participants' satisfaction levels across different explanations ($p = .056 > \alpha$) (see Table 5.5). In other words, the explanations of *Type 2* did not increase the *satisfaction* of the participants with regard to group recommendations. This also proves that the hypothesis H_1 can not be confirmed for the *ADD-based* explanation of *Type 2*. In contrast, in *Type 1* and *Type 3*, the Kruskal-Wallis tests reveal statistically significant differences in the *perceived fairness, perceived consensus,* and *satisfaction* levels

[‡] All the tests presented in this chapter were performed in the SPSS V.22.

[§] In hypothesis testing, a *Type I error* involves rejecting the null hypothesis (e.g., there are no differences among the groups) when it is actually true (Pallant, 2007).

across different explanations (see Table 5.5). The Mann-Whitney U tests show that these differences were triggered by *MPL-based* and *FAI-based* explanations (see Tables 5.6, 5.7, and 5.8).

Explanation types	*fairness*	*consensus*	*satisfaction*
Type 1	**.000**	**.000**	**.000**
Type 2	**.005**	**.000**	.056
Type 3	**.000**	**.000**	**.000**

Table 5.5.: *p values* (*2-tailed*) of the Kruskal-Wallis tests in the *perceived fairness, perceived consensus,* and *satisfaction levels* across different explanations in *Type 1, Type 2,* and *Type 3*.

Explanations	*fairness*	*consensus*	*satisfaction*
MPL vs. ADD	**.000**	**.000**	**.000**
MPL vs. APP	**.000**	**.000**	**.000**
MPL vs. LMS	**.000**	**.000**	**.000**
MPL vs. MAJ	**.000**	**.000**	**.000**

Table 5.6.: *p values* (*2-tailed*) of the Mann-Whitney U tests ($\alpha' = .005$) between the *MPL-based* explanation and one of the remaining explanations in *Type 1*.

Explanations	*fairness*	*consensus*	*satisfaction*
MPL vs. ADD	**.000**	**.000**	**.000**
MPL vs. APP	**.000**	**.000**	.005
MPL vs. LMS	**.000**	**.000**	.097
MPL vs. MAJ	**.000**	**.000**	**.000**

Table 5.7.: *p values* (*2-tailed*) of the Mann-Whitney U tests ($\alpha' = .003$) between the *MPL-based* explanation and one of the remaining explanations in *Type 3*.

Explanations	*fairness*	*consensus*	*satisfaction*
FAI vs. ADD	**.000**	.016	**.000**
FAI vs. APP	**.000**	.071	**.003**
FAI vs. LMS	**.001**	.085	.116
FAI vs. MAJ	**.000**	.005	**.000**
FAI vs. MAJ	.022	**.000**	.703

Table 5.8.: *p values* (*2-tailed*) of the Mann-Whitney U tests ($\alpha' = .003$) between the *FAI-based* explanation and one of the remaining explanations in *Type 3*.

To specify the best explanation, we inspected the mean ranks of the explanations of *Type 1* and *Type 3*. We found out that the *ADD-based* explanations achieved the highest perceived fairness and perceived consensus levels in *Type 1* and the highest satisfaction levels in *Type 3*. The *MAJ-based* explanations received the highest satisfaction levels in *Type 1* and the highest perceived fairness and perceived consensus in *Type 3* (see Table 5.9). We also recognized that in some dimensions, the mean ranks of these explanations are quite close to each other. Besides, the *APP-based* explanation also performed well and in some dimensions, its mean ranks are quite close to those of the *ADD-based* and *MAJ-based* explanations (see Table 5.9). Thereby, it could be the case that the *ADD-based*, *APP-based*, *MAJ-based* are the best

Explanations		*fairness*	*consensus*	*satisfaction*
Type 1	*ADD*	**143.61**	**143.51**	132.15
	APP	136.24	125.94	130.79
	MAJ	131.50	131.83	**132.47**
	LMS	114.85	118.68	117.72
	MPL	57.16	63.49	70.07
Type 3	*ADD*	182.90	174.67	**183.82**
	APP	184.68	165.54	167.87
	MAJ	**185.89**	**181.24**	182.08
	LMS	164.28	163.72	144.13
	MPL	83.24	80.92	115.32
	FAI	113.90	136.53	119.24

Table 5.9.: *Mean ranks* generated in the Kruskal-Wallis tests for all explanations in *Type 1* and *Type 3*.

Explanations		*fairness*	*consensus*	*satisfaction*
Type 1 ($\alpha' = .005$)	*ADD vs. APP*	.547	.134	.897
	ADD vs. MAJ	.329	.340	1.000
	APP vs. MAJ	.712	.638	.908
Type 3 ($\alpha' = .003$)	*ADD vs. APP*	.804	.615	.403
	ADD vs. MAJ	.736	.666	.870
	APP vs. MAJ	.954	.350	.497

Table 5.10.: *p values* (*2-tailed*) achieved from the Mann-Whitney U tests between pairs of *ADD-based, APP-based*, and *MAJ-based* explanations in *Type 1* and *Type 3*.

explanations. Indeed, by performing Mann-Whitney U tests between pairs of these explanations, we found out that among these explanations, there were no statistically significant differences in the participants' evaluations regarding the mentioned dimensions (see Table 5.10). That means, *ADD-based, APP-based*, and *MAJ-based* explanations best helped to increase the fairness perception, consensus perception, and satisfaction of the participants with group recommendations.

Discussion: An *ADD-based* explanation exposes a group recommendation strategy taking into account as far as possible the preferences of *all* group members, whereas other explanations describe the strategies considering the preferences of a *subset* of group members. Thereby, this explanation can convince the participants that the group recommendation is a *fairness-oriented solution*. Besides, a recommendation considering the preferences of all individuals would create a *consensus* among group members (Senot et al., 2010). This could explain as to why these explanations achieved high perceived consensus levels. Moreover, taking into account all group members' preferences is the premise of a more favorable recommendation. Therefore, this might lead to higher satisfaction levels of the participants with regard to the group recommendation.

APP-based and *MAJ-based* explanations are also considered as the best explanations. These explanations describe one of the most effective decision making techniques, so-called *majority rule*, which takes most group members' preferences into account (Hastie and Kameda, 2005). Therefore, these explanations helped to increase the fairness and consensus perceptions of the participants, although sometimes they were not so fair as the *ADD-based* explanation (evidently, the decision generated by the majority rule might be unfair to someone who is not in favor of the decision). Finally, *APP-based* and *MAJ-based* explanations also helped to increase the participants' satisfaction with regard to the group recommendation. This could be explained by the fact that when the fairness and consensus perceptions of the participants increase, so does the satisfaction of participants concerning the group recommendation (this can be confirmed in the hypothesis H_2 - see *Section 5.6.2*).

In conclusion, hypothesis H_1 can be confirmed for the *ADD-based* explanations of *Type 1* and *Type 3*. These explanations describe a group recommendation strategy considering the preferences of all individual group members. Besides, the explanations describing the *majority rule* also effectively help to increase the fairness perception, consensus perception, and satisfaction of users with regard to group recommendations.

5.6.2. Hypothesis H_2

(H_2 - *"There exist positive correlations between the perceived fairness (or the perceived consensus) of the explanations and the satisfaction of users with regard to group recommendations"*).

Data analysis method: For the hypothesis H_2, we examined two relationships: (1) *between perceived fairness and satisfaction* and (2) *between perceived consensus and satisfaction*. To address those, we collected and analyzed the participants' evaluations as follows:

- **Step 1**: For each explanation, we collected three evaluation sets corresponding to the three mentioned dimensions.

- **Step 2**: We performed two *Spearman Rank Order Correlation* tests ($\alpha = .05$). These tests investigate the *direction* and the *strength* of a monotonic relationship based on a *correlation coefficient r*. Regarding the *direction* of a relationship, $r > 0$ indicates a *positive correlation* (i.e., *as one variable increases, so does the other*). In contrast, $r < 0$ indicates a *negative correlation* (i.e., *as one variable increases, the other decreases*) Pallant (2007). Concerning the *strength* of a relationship, Cohen (1988) suggests that a relationship is *'weak'* if *r is from .10 to .29*, *'moderate'* if *r is from .30 to .49*, and *'strong'* if *r is from .50 to 1.0*.

Results and discussion:

- Between *perceived fairness* and *satisfaction*: The Spearman Rank Order Correlations summarized in Table 5.11 show that, in most explanations, there existed *positive correlations* between these two dimensions ($r > 0$). Furthermore, the perceived fairness levels revealed *moderate* or *strong* correlations with the satisfaction levels. One exception was detected in the *ADD-based* explanation of *Type 1*. In this explanation, the Spearman Rank Order Correlation test points out that there was not enough evidence to ascertain a relationship between the perceived fairness levels and the satisfaction levels of the participants with regard to the group recommendation (*p value* = *.051* $> \alpha$).

- Between *perceived consensus* and *satisfaction*: In most explanations, Spearman Rank Order Correlation tests reveal *positive correlations* between perceived consensus levels and satisfaction levels (see Table 5.11). In addition, an inspection of the correlation coefficients suggests that there were *moderate* or *strong* relationships between these two dimensions. However, in *ADD-based* and *LMS-based* explanations of *Type 2*, the Spearman Rank Order Correlation tests show that there was not enough evidence to confirm the correlations between the perceived consensus levels and the satisfaction levels of the participants with regard to the group recommendation (*p values* $> \alpha$).

In conclusion, the hypothesis H_2 can be wholly confirmed for all explanations of *Type 3*. That means, in the explanations including future decision plans, *higher perceived fairness (or perceived consensus) levels of explanations associate with higher satisfaction levels of users with regard to group recommendations*.

Explanations		*fairness vs. satisfaction*		*consensus vs. satisfaction*	
		r	*p(2-tailed)*	*r*	*p(2-tailed)*
Type 1	*ADD*	.290	.051	.466**	**.001**
	APP	.444**	**.002**	.741**	**.000**
	LMS	.562**	**.000**	.461**	**.001**
	MAJ	.581**	**.000**	.421**	**.004**
	MPL	.794**	**.000**	.748**	**.000**
Type 2	*ADD*	.538**	**.000**	.226	.126
	FAI	.390**	**.001**	.500**	**.000**
	APP	.342*	**.019**	.326*	**.026**
	LMS	.356*	**.014**	.242	.101
	MAJ	.616**	**.000**	.523**	**.000**
	MPL	.623**	**.000**	.440**	**.002**
Type 3	*ADD*	.570**	**.000**	.473**	**.001**
	FAI	.457**	**.000**	.373**	**.001**
	APP	.300*	**.046**	.510**	**.000**
	LMS	.815**	**.000**	.642**	**.000**
	MAJ	.441**	**.002**	.468**	**.001**
	MPL	.515**	**.000**	.716**	**.000**

Correlation is significant at the .05 level (2-tailed)
**Correlation is significant at the .01 level (2-tailed)*

Table 5.11.: Spearman Rank Order Correlations between the *perceived fairness/perceived consensus levels* and *satisfaction levels* of all explanations in *Type 1, Type 2,* and *Type 3*.

5.6.3. Hypothesis H_3

(H_3 - *"In the context of repeated decisions, the integration of the information regarding group members' satisfaction from previous decisions and future decision plans into social choice-based explanations is assumed to increase the fairness and consensus perception of users with regard to group recommendations"*.)

Data Analysis Method: For the hypothesis H_3, we tried to examine whether the explanations of *Type 2* and *Type 3* are better than those of *Type 1* in terms of increasing *perceived fairness* and *perceived consensus*. To address this hypothesis, we collected and analyzed the participants' evaluations according to the following steps:

- **Step 1**: For each explanation in each type, we collected two sets of evaluations corresponding to the two mentioned dimensions. As mentioned in the hypothesis H_1, these evaluations are *ordinal variables*, *independent*, and *not normally distributed*.

- **Step 2**: In each dimension of a specific explanation, we ran two Mann-Whitney U tests ($\alpha = .05$): (1) between *Type 1* and *Type 2* and (2) between *Type 1* and *Type 3*. To control the *Type I errors*, we applied a *Bonferroni adjustment* (Pallant, 2007) to revise the significance level ($\alpha' = \alpha/2 = .025$). In case these tests revealed statistically significant differences, we further inspected the mean ranks generated in these Mann-Whitney U tests to find out whether the explanations of *Type 2* and *Type 3* performed better than those of *Type 1*. We want to highlight that, when testing this hypothesis, we did not analyze the *FAI-based* explanation since this explanation only exists in *Type 2* and *Type 3*, but not in *Type 1*.

Results:

- *Between Type 1 and Type 2*: Mann-Whitney U tests show that there were no statistically significant differences in the perceived fairness and perceived consensus levels across *APP-based*, *LMS-based*, and *MAJ-based* explanations (see Table 5.12). In contrast, in the *ADD-based* explanation, we found a statistically significant difference regarding the perceived consensus ($p = .015 < \alpha'$). However, the mean ranks in Table 5.12 show that the perceived consensus levels of this explanation in *Type 2* were lower than those in *Type 1*. This means, the inclusion of the information of disadvantaged users from previous decisions did not improve the perceived consensus of the *ADD-based* explanation. However, in the *MPL-based* explanation, such an inclusion significantly improved the participants' fairness perception ($p = .013 < \alpha'$). In fact, the mean ranks in Table 5.12 obviously show that the participants provided *higher fairness-related evaluations* for this explanation of *Type 2*.

Explanations	*fairness*			*consensus*		
	p (*2-tailed*)	*mean rank* *Type 1*	*mean rank* *Type 2*	*p* (*2-tailed*)	*mean rank* *Type 1*	*mean rank* *Type 2*
ADD	.143	50.97	43.12	**.015**	53.49	40.65
APP	.155	51.37	43.63	.811	48.14	46.86
LMS	.713	45.99	47.99	.968	46.89	47.11
MAJ	.810	47.65	46.36	.306	49.77	44.29
MPL	**.013**	40.79	54.21	.103	43.05	51.95

Table 5.12.: Mann-Whitney U tests ($\alpha' = .025$) for all explanations between *Type 1* and *Type 2*.

- *Between Type 1 and Type 3*: On both fairness and consensus dimensions, the Mann-Whitney U tests do not reveal any statistically significant difference in the participants' evaluations for all explanations between *Type 1* and *Type 3* (*all p values* $> \alpha'$) (see Table 5.13). This means, the integration of future decision plans into the explanations did not increase the fairness and consensus perceptions of the participants with regard to group recommendations.

Explanations	*fairness*			*consensus*		
	p (*2-tailed*)	*mean rank* *Type 1*	*mean rank* *Type 3*	*p* (*2-tailed*)	*mean rank* *Type 1*	*mean rank* *Type 3*
ADD	.301	49.25	43.75	.460	48.46	44.54
APP	.919	46.77	46.22	.912	46.21	46.80
LMS	.797	45.80	47.20	.731	45.58	47.42
MAJ	.732	45.59	47.41	.534	44.85	48.15
MPL	.900	46.82	46.17	.779	45.77	47.27

Table 5.13.: Mann-Whitney U tests ($\alpha' = .025$) for all explanations between *Type 1* and *Type 3*.

Discussion: The results can be explained as follows. The *MPL-based* explanation describes a preference aggregation strategy that only supports group members who provided the highest ratings for items. In other words, this strategy recommends an item on the basis of taking into account the preferences of *a subset of group members*. This results in the dissatisfaction of some group members who provided lower ratings for the recommended item.

In the proposed group decision scenario (see Table 5.4), the *MPL* aggregation strategy recommends the *Rest C* to the group since it achieves the highest of all group members' ratings. It seems that this decision only supports the preference of Alex and ignores others' preferences. This causes the dissatisfaction of Anna, Sam, and Leo. However, in such a situation, if a *MPL-based* explanation of *Type 2* is provided (e.g., *"Rest C has been recommended to the group since it achieves the highest of all group members' ratings.* ***This decision supports the preference of Alex, who was treated less favorably in the last three decisions****"*), then group members might be aware of the fairness of the group recommendation. This explanation reminds group members of the disadvantaged person (Alex) in previous decisions. Thereby, they might accept that Alex's preferences would have a higher priority in the on-going decision.

In our user study, each participant might assume his/herself as a group member to better perceive the inequitable situation of Alex. The participant might apply the *Equity Theory* (Tanner, 2018) to analyze the fairness aspect of the explanations. According to this theory, a situation is *equitable* when users who invested similar efforts should receive similar rewards. However, this seems not to be the case in the mentioned scenario. Besides, according to the *Equity Theory*, a user who is aware of an *inequitable treatment* will be emotionally motivated to gain equity. Thanks to the *MPL-based* explanation, the participant was aware of the inequity that occurred inside the group. Therefore, he/she gave a high evaluation for the perceived fairness of this explanation.

In conclusion, in the context of repeated decisions, the hypothesis H_3 can be only confirmed for the *perceived fairness* of the *MPL-based* explanation of *Type 2*. In other words, only the information about less-favored group members from previous decisions helps to significantly increase users' fairness perception about group recommendations. Such information is only helpful for the explanations which describe a group recommendation strategy taking into account the preferences of a subset of group members.

5.7. Conclusion and Future work

In this chapter, we proposed three explanation types taking into account group dynamics aspects, such as *fairness, consensus*, and *satisfaction* of users with regard to group recommendations. We found out that the explanations convey group recommendation strategies taking into account preferences of *all* or *a majority* of group members perform the best in terms of the mentioned aspects. Moreover, we also discovered *positive correlations* between the perceived fairness levels of explanations and the satisfaction levels of users with group recommendations, especially for the explanations including future decision plans. In these explanations, higher perceived fairness levels of explanations associate with higher satisfaction levels of users with regard to group recommendations. The same correlations hold between the perceived consensus levels and the satisfaction levels. Furthermore, in the context of repeated decisions, the inclusion of group members' satisfaction from previous decisions can help to improve the fairness perception of users about group recommendations. Such an inclusion is particularly helpful for the explanations which describe a group recommendation strategy taking into account preferences of a subset of group members.

One limitation of the paper lies in the distribution of explanations to the participants. As mentioned in *Section 5.5.2*, each participant had to evaluate a sequence of six different explanations. At any given time, each participant could observe and evaluate *only one* explanation, and the evaluation for one explanation was independent of the evaluations for other explanations. However, since the break time between two different evaluations was not long enough, this could trigger potential biases in the evaluation process of the participants. As a result, the participants' evaluations for the explanations were not wholly independent of each other. For addressing such a potential issue, in the future, we will run our data with *multi-level models* to achieve more precise results.

On the other hand, within the scope of future work, we will extend this study by investigating fairness and consensus aspects in the context of *different decision domains*. An example explanation in such a context could be: *"The preferences of user X were not taken into account in the last two decisions - restaurants* *and movies decisions. Therefore, this user will have a higher priority in upcoming decisions regarding restaurants and movies"*. Besides, we will integrate other criteria into social-choice based explanations to investigate their impacts on the perceived fairness and perceived consensus of group recommendations. Some examples of these criteria could be: (*i*) group members' personality (e.g., *"a group member with a strong personality who was treated unfavorably last time will be immediately compensated in the next decision"*), (*ii*) group members' personal situations (e.g., *"user u_a is a vegan, therefore Restaurant A has been chosen to the group since this restaurant additionally serves vegan dishes"*), or (*iii*) item domains (e.g., an example explanation is formulated for the *high-involvement item domain* as follows: *"Although the apartment X is not the favorite option of all members, no group member has a real problem with it. Therefore, this apartment seems to be the most appropriate solution for all group members to stay together in the next two years"*).

Chapter 6

User Interfaces for Counteracting Decision Manipulation In Group Recommender Systems

This chapter is based on the results documented in (Tran et al., 2019b). All parts of this chapter regarding literature research, proposing user interfaces, user study, data analysis, and writing the whole chapter were done by the author of the thesis.

6.1. Abstract

In group recommender systems, a *decision manipulation* refers to an attack in which a group member tries to adapt the rating of items in order to push his/her favorite options. This action results in serious vulnerabilities to the quality of group recommendations. In this context, making rating adaptations of group members transparent is assumed to counteract decision manipulation issues. Based on this assumption, we proposed user interfaces visualizing information dimensions regarding rating adaptations of group members at different transparency levels. We conducted a user study to investigate user interfaces (UIs) which are the most understandable and effectively help to counteract decision manipulation. The results show that the UI at the highest transparency level best helps to discourage users from decision manipulation. We also figured out that at a specific transparency level, the ability of the UIs to counteract decision manipulation differs according to the information dimensions represented in the UIs. The UIs showing group members *who have adapted the rating of items* better help to counteract decision manipulation compared to those which do not include this information. In addition, the information dimensions of *'item ratings'* and *'group recommendations'* have the strongest impacts on preventing users from decision manipulation.

6.2. Introduction

Recommender systems have been recognized as being useful tools in modern e-commerce. They guide users to items they might love based on their own and other users' preferences (Resnick and Sami, 2008). However, recommender systems supporting *explicit user feedback* (Herlocker et al., 1999; Lieberman et al., 1999b; Ekstrand et al., 2011) might face biases triggered by '*insincere*' user preferences. In this scenario, users try to change their feedback to alter the recommendation to a preferred one (Herlocker et al., 2004; Jawaheer et al., 2014). Such *decision manipulations* can deteriorate the quality of recommendations and decrease the trust of users in recommender systems (Chirita et al., 2005).

In *single-user recommender systems*, one of the most popular manipulation scenarios can be found in *collaborative filtering* recommender systems where a vendor creates many online identities and uses each of them to strategically rate items. He/she defines high ratings for his/her products and low ratings for competitors' products (Van Roy and Yan, 2009). This action is referred to as a *shilling attack* in which the attacker gives biased ratings to affect the recommendation (Li and Luo, 2011; Zhou et al., 2015).

In *group recommender systems* that support group decision making processes, manipulation issues can also be triggered when user control mechanisms (McNee et al., 2003; Xiao and Benbasat, 2007; Jannach et al., 2017) are implemented. Some user controls supported in group recommender systems thereof can be: *(i)* to allow users to articulate their preferences for items, *(ii)* to enable users to see others' preferences, or *(iii)* to adapt their preferences for achieving a consensus among group members (McCarthy et al., 2006; Palomares et al., 2014a; Stettinger et al., 2015). These mechanisms facilitate group members' rating adaptations, which aim to push their favorite options (Conitzer and Yokoo, 2010; Jameson et al., 2003). In other words, these actions of group members aim to gain their interests themselves rather than the whole group's interest.

Up to now, to some extent, it is still unclear how to counteract decision manipulation in group recommender systems. Although there are some studies on manipulation resistance (Chirita et al., 2005; Van Roy and Yan, 2009; Li and Luo, 2011; Gunes et al., 2014; Zhou et al., 2015), most of the proposed solutions are for single-user scenarios. To the best of our knowledge, in the current literature, there exist only a few research contributions with an in-depth analysis of decision manipulation issues (Jameson et al., 2003, 2004; Lang et al., 2010). In this chapter, we propose UIs for counteracting decision manipulation in group recommender systems. Our idea is to '*disclose*' the rating adaptation history of group members, which means all the past item rating changes of group members are reported and shown to the whole group. The rating adaptation history could consist of the following information: *group members* who have adapted the rating of items, *items* whose ratings have been adapted, *ratings* of items, *timeline* of rating adaptations, and *changes of group recommendation* during the rating adaptation process. We assume that making group members' rating adaptations *transparent* can help to discourage users from decision manipulation.

The contribution of this chapter is threefold:

1. We propose UIs which visualize the rating adaptation history of group members at different transparency levels.
2. We identify UIs which are the most understandable and most effective for counteracting decision manipulation in group recommender systems.
3. We discuss information dimensions of rating adaptations which strongly help to prevent users from decision manipulation.

The remainder of the chapter is organized as follows. In *Section 6.3*, we summarize the related work of decision manipulation issues in group recommender systems. In *Section 6.4*, we present different dimensions describing the rating adaptation history of group members and propose UIs on the basis of combining these dimensions. In *Section 6.5*, we define research questions and describe main steps of our user study. Data analysis and results regarding the research questions are presented in *Section 6.6*. In *Section 6.7*, we conclude the chapter and discuss open issues for future work.

6.3. Related Work

Decision manipulation has been experienced in one of the earliest group recommender systems, so-called MUSICFX (McCarthy and Anagnost, 1998). This system automatically selects music genres to play in a fitness center. When using this system, some users were observed to intentionally indicate that they disliked the being-played genre in order to force an immediate change of music genre, even if they just liked it a bit less than other genres. Another manipulation behavior was found in the TRAVEL DECISION FORUM (Jameson et al., 2004). In this system, group members can see the preferences of each other. This could trigger rating adaptations of some group members for pushing their favorite options. In such a situation, if users do not know each others' preferences, it would be more challenging to manipulate the decision. Therefore, to counteract this, the preferences of group members should not be shown in the preference articulation phase (Stettinger et al., 2015). However, this seems not to be an optimal solution since users might be able to guess others' preferences (at least, roughly) (Jameson and Smyth, 2007), and in some cases, group members' preferences have to be shown (e.g., for consensus making purpose).

An alternative to counteract decision manipulation is to have a group recommender system applying a preference aggregation strategy* that is inherently *non-manipulatable* (Jameson et al., 2004). One simple non-manipulatable aggregation strategy is *'median'* which takes the item falling exactly in the middle of an ordered list of group members' preferences. Another strategy is *'random choice'* that randomly selects an item from a given item list (Jameson et al., 2004). However, these strategies result in sub-optimal recommendations which reveal the unacceptability of users with regard to group recommendations. In the line of *mechanism design* research, Conitzer and Sandholm (2002); Conitzer and Yokoo (2010); Sandholm (2016) proposed other approaches that automatically generate aggregation functions so that desirable recommendations can be achieved for groups, even if group members rate items based on their self-interest. However, these approaches face the difficulty in providing understandable and adequate explanations of group recommendations.

In this chapter, we propose a UI approach to *showing group members' rating adaptations* to the whole group. The idea is that if a user (as a manipulator) knows that others can see his/her rating adaptations, then he/she might not try to manipulate the decision. This originates from an observer effect, so-called *Hawthorne effect*. *"The Hawthorne effect is a type of reactivity in which users modify an aspect of their behavior in response to their awareness of being observed"* (Sedgwick and Greenwood, 2015). This effect indicates a psychological phenomenon in which users tend to do something positive or better if they are aware of being observed by others. In the context of decision manipulation, this effect can be interpreted by the fact that users tend to avoid decision manipulations if they know that others can see their behaviors.

6.4. User Interfaces for Counteracting Decision Manipulation

Before designing UIs, we propose some information dimensions which describe the rating adaptations of group members. These dimensions are included in the proposed UIs.

6.4.1. Rating Adaptation Information

The rating adaptations of group members could be made for either *positive* purposes (e.g., making consensus among group members) or *negative* purposes (e.g., manipulating the decision). In this context, a paramount concern is that *"which information should be shown so that negative-purpose rating adaptations of group members can be detected"*. We assume only showing the information regarding *"which of items whose ratings have been adapted by group members"* is insufficient to predict group members' decision manipulation behaviors. Additional information needs to be clarified, such as *"how the ratings of these items have been adapted"*, *"when these ratings were adapted"*, and *"how group recommendation*

*In group recommender systems, a preference aggregation strategy is applied to merge all individual group members' preferences and to generate a group recommendation (Felfernig et al., 2018a).

has been changed after rating adaptations". To address this, we propose the following dimensions:

- *Dimension 1 (Group member - GM)* indicating the *name* of a group member who has adapted the ratings of items;
- *Dimension 2 (Item - I)* indicating the item whose rating has been adapted;
- *Dimension 3 (Rating - R)* describing the rating of an item which can be represented in three forms:
  - *Original rating*: The rating is given by the group member at the starting point of the preference articulation phase.
  - *Adapted rating*: The rating that has been adapted by the group member.
  - *Group rating*: The rating calculated by merging all individual group members' ratings using a preference aggregation strategy (Masthoff, 2011; Felfernig et al., 2018b).
- *Dimension 4 (Timeline - TL)* displaying group members' rating adaptations in a chronological order. This dimension indicates *when* a group member adapted the ratings of items and *how often* of his/her rating adaptations. A manipulator might adapt the ratings of items many times until the system chooses his/her favorite item. This dimension brings an additional clue to detect the decision manipulation behavior of group members.
- *Dimension 5 (Tendency - TD)* describing the *direction* (*increase* (+) or *decrease* (-)) and the *magnitude* of a rating adaptation. For instance, for pushing the group recommendation to an item *X*, the manipulator first increases the rating of this item by 2 points (+2) and then decreases the ratings of another item by 3 points (-3). This dimension can be beneficial for detecting the decision manipulation attempts of group members.
- *Dimension 6 (Group recommendation - GR)* revealing how the group recommendation has been changed according to the rating adaptations of group members. For this dimension, in the rating adaptation history we show the *group recommendation at the starting point* (i.e., the group recommendation generated when all group members have just rated for the items) and the *group recommendation after each adaptation* (i.e., the group recommendation generated when a group member has adapted the rating of an item). This dimension can be beneficial to detect decision manipulation attempts of group members. It is more likely that the rating adaptations of manipulators lead to group recommendation changes.

6.4.2. User Interfaces for Counteracting Decision Manipulation

The UIs for counteracting decision manipulation were generated and visualized in the following steps:

Step 1 - Generate the UIs: In this step, we generated two groups of UIs (*Group 1* and *Group 2*) by combining the dimensions presented in *Section 6.4.1*. These UIs represent the rating adaptation history of group members at different transparency levels. The *transparency level* of a UI corresponds to *the number of dimensions* included in the UI.

In *Group 1*, each UI shows at least the information of *"who has adapted the ratings of which items"*. Thereby, each UI at least consists of the two following dimensions: *"Group member"* and *"Item"*. A basic UI (named $UI1_{basis}$) can be generated by these two dimensions with the transparency level of 2. To design other UIs with higher transparency levels, we gradually add the remaining dimensions to the $UI1_{basis}$. The construction of the UIs of *Group 1* is described in Table 6.1. In total, there are *16* UIs for *Group 1*

Transparency level	UI	*GM*	*I*	*R*	*TL*	*TD*	*GR*
2	$UI1_{basis}$	√	√				
3	$UI1_{R}$	√	√	√			
	$UI1_{TL}$	√	√		√		
	$UI1_{TD}$	√	√			√	
	$UI1_{GR}$	√	√				√
4	$UI1_{R+TL}$	√	√	√	√		
	$UI1_{R+TD}$	√	√	√		√	
	$UI1_{R+GR}$	√	√	√			√
	$UI1_{TL+TD}$	√	√		√	√	
	$UI1_{TL+GR}$	√	√		√		√
	$UI1_{TD+GR}$	√	√			√	√
5	$UI1_{R+TL+TD}$	√	√	√	√	√	
	$UI1_{R+TL+GR}$	√	√	√	√		√
	$UI1_{R+TD+GR}$	√	√	√		√	√
	$UI1_{TL+TD+GR}$	√	√		√	√	√
6	$UI1_{all}$	√	√	√	√	√	√

Table 6.1.: The construction of the UIs of *Group 1*, where *GM - Group member, I - Item, R - Rating, TL - Timeline, TD - Tendency,* and *GR - Group recommendation*. The ratings represented in the UIs of *Group 1* are *original ratings* and *adapted ratings*.

Transparency level	UI	*I*	*TL*	*R*	*TD*	*GR*
3	$UI2_{R}$	√	√	√		
	$UI2_{TD}$	√	√		√	
	$UI2_{GR}$	√	√			√
4	$UI2_{R+TD}$	√	√	√	√	
	$UI2_{R+GR}$	√	√	√		√
	$UI2_{TD+GR}$	√	√		√	√
5	$UI2_{all}$	√	√	√	√	√

Table 6.2.: The construction of the UIs of *Group 2* where *I - Item, TL - Timeline, R - Rating, TD - Tendency,* and *GR - Group recommendation*. The ratings represented in the UIs of *Group 2* are *group ratings*.

categorized into *five* transparency levels ranging from 2 to 6.

We assumed the UIs of *Group 1* could effectively help to counteract decision manipulation. However, showing the *"Group member"* dimension could raise *privacy* issues. Therefore, we additionally proposed *Group 2* whose UIs do not include the *'Group member'* dimension. In this chapter, we investigate which group of UIs performs better in terms of decision manipulation counteraction. The UIs of *Group 2* do not show the rating adaptations of each group member. Instead, they reveal how the *group ratings* of items have been changed by time. Thereby, each UI always include the two following dimensions: *"Item"* and *"Timeline"*. The UIs of *Group 2* are designed by gradually combining these two dimensions with the remaining dimensions. The construction of the UIs of *Group 2* is described in Table 6.2. In total, there are *seven* UIs categorized into *three* transparency levels ranging from 3 to 5.

Step 2 - Select the UIs: In this step, we inspected the UIs generated in *Step 1* and got rid of some UIs which are not understandable or represent similar information with other UIs. For instance, in *Group 1*, we recognized that $UI1_{R+TL}$ and $UI1_{R+TL+TD}$ represent almost the same information. Indeed, compared

Transparency level	Group	UI	Dimensions	Visualization method
2	1	$UI1_{basis}$	GM, I	*table* (Figure 6.1)
3	1	$UI1_{R}$	GM, I, R	*table*
	1	$UI1_{TL}$	GM, I, TL	*table*
	1	$UI1_{TD}$	GM, I, TD	*table*
	1	$UI1_{GR}$	GM, I, GR	*table* (Figure 6.2)
	2	$UI2_{R}$	I, TL, R	*table*
	2	$UI2_{TD}$	I, TL, TD	*table*
4	1	$UI1_{R+TD}$	GM, I, R, TD	*graph* (Figure 6.3)
	1	$UI1_{TL+TD}$	GM, I, TL, TD	graph
	1	$UI1_{TL+GR}$	GM, I, TL, GR	*table*
	1	$UI1_{TD+GR}$	GM, I, TD, GR	*table*
	2	$UI2_{TD+GR}$	I, TL, TD, GR	*table*
5	1	$UI1_{R+TL+TD}$	GM, I, R, TL, TD	*graph*
	1	$UI1_{R+TD+GR}$	GM, I, R, TD, GR	*graph*
	1	$UI1_{TL+TD+GR}$	GM, I, TL, TD, GR	*table* (Figure 6.4)
	2	$UI2_{all}$	I, TL, R, TD, GR	*table* (Figure 6.5)
6	1	$UI1_{all}$	GM, I, R, TL, TD, GR	*graph* (Figure 6.6)

Table 6.3.: The UIs of *Group 1* and *Group 2* which are selected for the user study.

to the $UI1_{R+TL+TD}$, the $UI1_{R+TL}$ does not include the *"Tendency"* dimension. However, this information can be figured out based on the *"Rating"* dimension (i.e., if we know the initial rating and adapted rating of an item, then we can figure out how this item has been adapted). In this case, we decided to keep the $UI1_{R+TL+TD}$ and removed the $UI1_{R+TL}$ since with the *'Tendency'* dimension, the rating adaptation history of group members might be clearer and more understandable to users.

For the same reason, in *Group 1*, we removed $UI1_{R+GR}$ and $UI1_{R+TL+GR}$ because they respectively represent similar information to $UI1_{R+TD+GR}$ and $UI1_{all}$. In *Group 2*, we got rid of $UI2_{R+TD}$ and $UI2_{R+GR}$ since they represent similar information to $UI2_{R}$ and $UI2_{all}$ respectively. Besides, the $UI2_{GR}$ was omitted as well because it could be not so understandable to users. Consequently, we retained *17 UIs* for both groups which are categorized into *five* transparency levels as shown in Table 6.3. In the rest of the chapter, we only focus on analyzing these 17 UIs.

Step 3 - Visualize the UIs: Before visualizing the UIs, we defined a decision manipulation scenario in group recommender systems. In our study, this scenario was described in the context of group decisions where *group members know each other quite well* (e.g., friends or family members) and jointly decide on *a small given set of items* (e.g., 3-5 items). Since group members are familiar with each other, some group member's attempts to push his/her favorite options could result in negative impacts on the group decision, such as *fairness validation* and *cohesion damage* among group members. Therefore, avoiding decision manipulation in this context is crucial to conserve the decision quality as well as the group cohesion. Our decision manipulation scenario was described as follows:

"Suppose that a group of four friends (Alex, Maria, Rosie, and Thomas) used a group recommender system to decide on a tourism destination type for the upcoming holiday. The group members explicitly rated three destination types (*museum, sea,* and *mountain*) using a 5-star rating scale ranging from *1 (the worst)* to *5 (the best)* (see Table 6.4). After articulating their preferences for destination types, each member can see others' preferences. The system recommends to the group a destination type with the maximum

Group member	*museum*	*sea*	*mountain*
Rosie	3	3	4
Alex	2	2	2
Maria	1	4	3
Thomas	2	3	4
group rating	2	3	**3.25**

Table 6.4.: Ratings of group members for the destination types (1 = *the worst*, 5 = *the best*).

average of individuals' ratings. With the articulated ratings, the *"mountain"* is recommended to the group.

Alex does not like any destination type. However, he recognizes that most group members seem to be interested in *"mountain"*. Therefore, he increases his rating for this option *from 2 to 3 stars* to speed up the consensus achieving process within the group. The rating adaptation of Alex does not change the group recommendation. Maria likes *"sea"* and wants the system will choose this option. Therefore, she first increases the rating of this option (*from 4 up to 5 stars*) and then decreases the rating of the *"mountain"* (*from 3 down to 1 star*). The rating adaptations of Maria push the group recommendation to the *"sea"* option. The behavior of Maria in this context is a so-called *decision manipulation* and Maria is a *manipulator*."

Based on this scenario, we visualized the *17 selected UIs* described in Table 6.3. We inspected existing visualization methods in recommender systems (He et al., 2016) and selected *tables* and *graphs* methods to visualize the proposed UIs. The reason was that compared to other visualization methods, *tables* and *graphs* are quite appropriate to intuitively visualize different dimensions of rating adaptation history in a UI. After visualizing the UIs, we conducted a pilot user study with *eight* experts in our institute[†] who are working in the fields of Software Engineering and UI design to get feedback concerning the visualization method and the understandability of the UIs. After that, we collected the experts' feedback and improved the UIs. The visualization of the UIs are briefly presented as follows:[‡]

- *At the transparency level of 2,* the $UI1_{basis}$ is visualized using a *2-column table* on which the first column shows *the names of group members*, the second one represents *the items*, and each row indicates that *a group member has adapted the rating of an item.* The visualization of this UI is shown in Figure 6.1.
- *At the transparency level of 3*, all UIs are visualized using *multi-column tables*. The number of the columns of each table differs according to the dimensions integrated into the UI. Each table consists of columns which show *the names of group members* and *items*. The remaining columns are for representing additional dimensions. For instance, in the visualization of the $UI1_{GR}$ (see Figure 6.2), the two last columns show *group recommendations*. One column shows the group recommendations generated at the starting point. Another one shows the group recommendations generated at the time when group members adapted the ratings of items.
- *At the transparency level of 4*, the $UI1_{R+TD}$ (see Figure 6.3) is visualized using a graph where the Y-axis shows *rating values* of a 5-star rating scale and the X-axis shows *the names of group members*. The space between these two axes shows *the items* and the *tendency* of rating adaptations. The tendency is visualized using arrows with different directions and lengths. The direction of an arrow indicates how the rating of an item has been increased or decreased. The length of an arrow represents how much the rating of an item has been adapted. The $UI1_{TL+TD}$ is also visualized using

[†] Institute of Software Technology - Graz University of Technology, Graz, Austria
[‡] For the visualization of the proposed UIs, we refer to the link: http://www.ist.tugraz.at/trang/ManipulationCounteractionUIs/

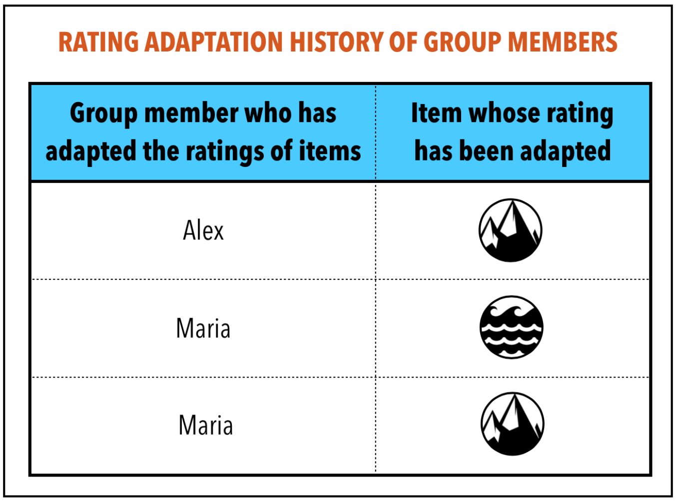


Figure 6.1.: The visualization of $UI1_{GR}$ (transparency level = 2). Each row indicates that *"a group member has adapted the rating of an item"*. For instance, the first row shows that: *"Alex has adapted the rating of the mountain option"*.

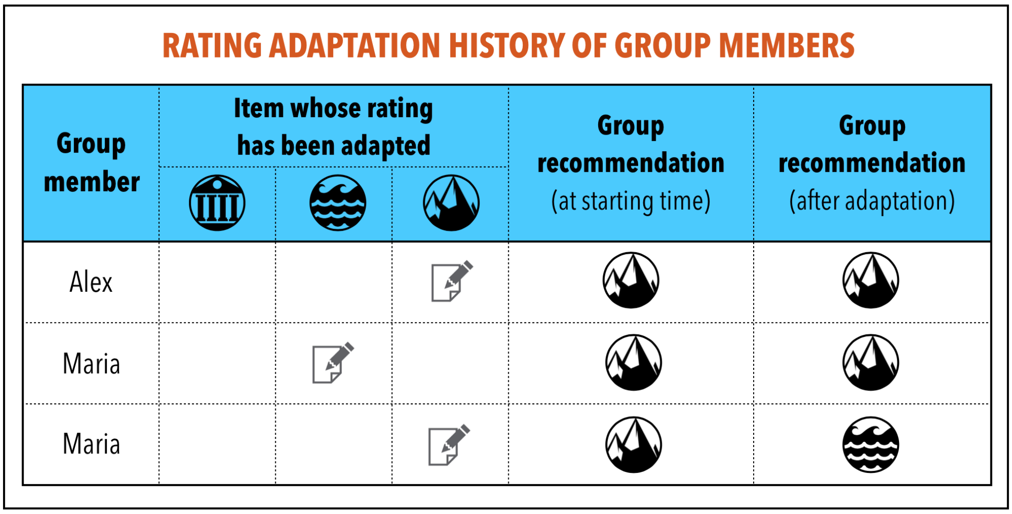


Figure 6.2.: The visualization of $UI1_{GR}$ (transparency level = 3). The blank cell indicates that a group member has not adapted the rating of an item. The *edit icon* indicates that a group member has adapted the rating of an item.

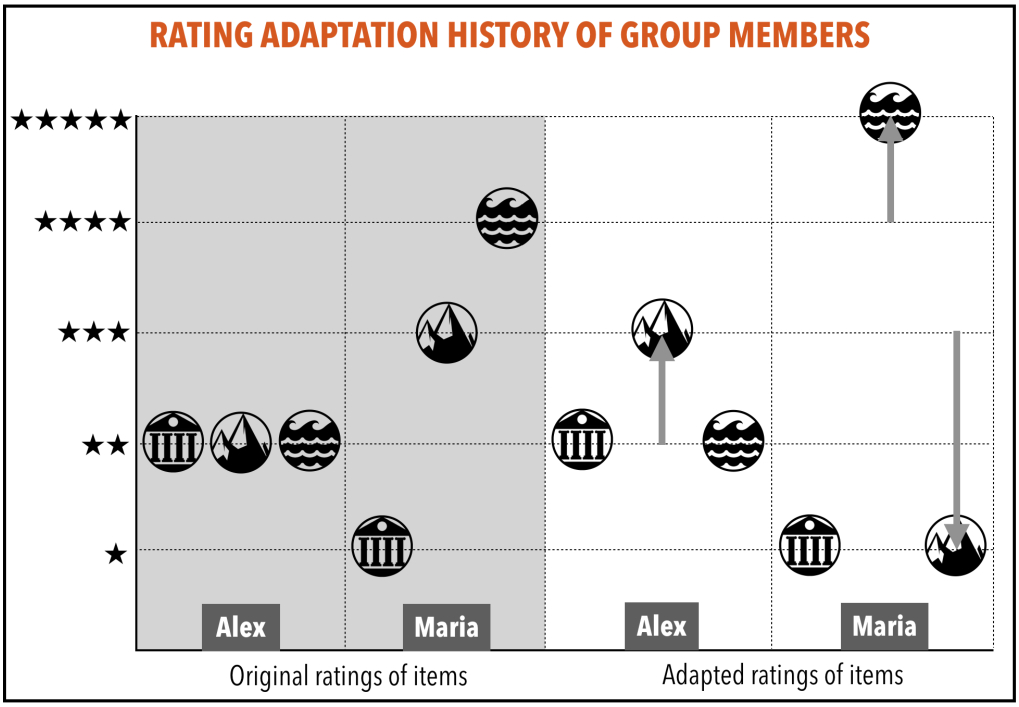


Figure 6.3.: The $UI1_{R+TD}$ (transparency level = 4) visualizing the dimensions of *"group member"*, *"item"*, *"rating"*, and *"tendency"*.

a graph on which the Y-axis shows *the names of items* and the X-axis shows a *timeline*. Between these two axes, the *tendency* of rating adaptations is visualized using directed arrows. The remaining UIs are visualized using *multi-column tables* whose structures are quite similar to the UIs visualized at the transparency level of 3.

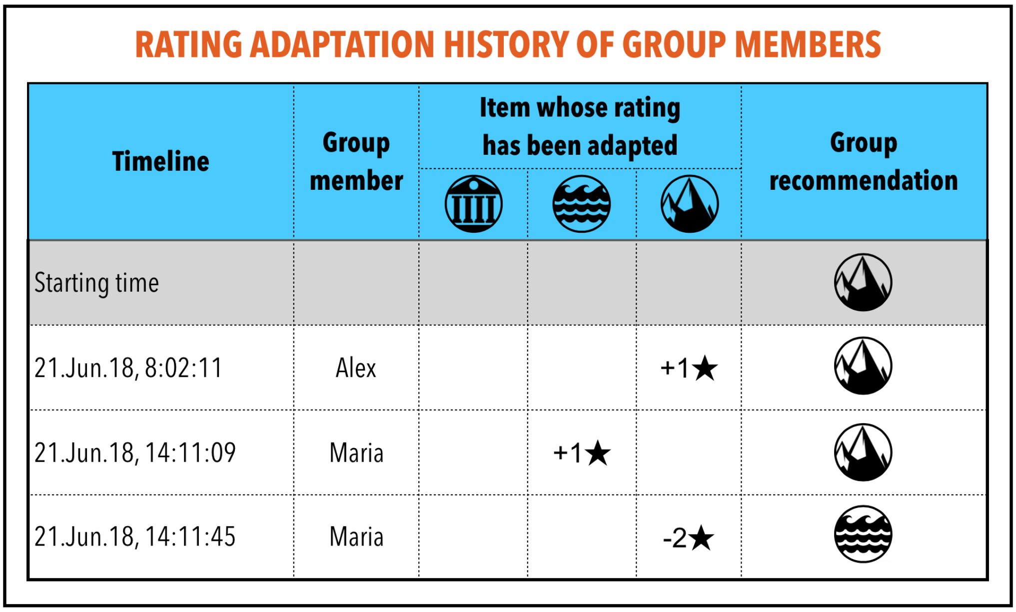
- *At the transparency level of 5*, the $UI1_{R+TL+TD}$ is visualized using a graph whose structure is similar to the one shown in the $UI1_{R+TD}$ (see Figure 6.3). However, in the X-axis, besides the names of group members, a *timeline* is also represented. For visualizing the $UI1_{R+TD+GR}$, the graph of the $UI1_{R+TD}$ is again used. In this graph, an additional axis representing the changes of group recommendations is inserted. Finally, the remaining UIs, $UI1_{TL+TD+GR}$ and $UI2_{all}$, are visualized using *multi-column tables* as respectively depicted in Figure 6.4 and Figure 6.5.
- *At the transparency level of 6*, the $UI1_{all}$ is visualized using a graph as shown in Figure 6.6.

6.5. Research Questions and User Study

In this section, we first define research questions to investigate UIs which are the most effective for counteracting decision manipulation in group recommender systems. After that, we conduct a user study in order to address the research questions.

6.5.1. Research Questions

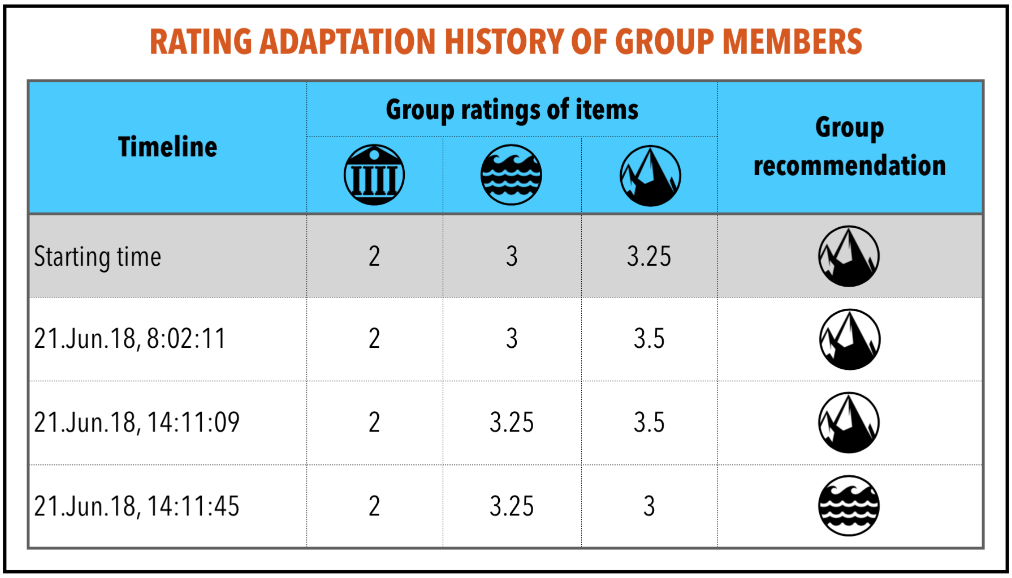
One of our goals is to find out at which transparency level of rating adaptation history, the UIs best help to counteract decision manipulation. We assume that *the higher the transparency level of the rating adaption history, the lower the preparedness of users concerning decision manipulation*. Thereby, the UIs at the



RATING ADAPTATION HISTORY OF GROUP MEMBERS

Timeline	Group member	Item whose rating has been adapted			Group recommendation
Starting time					
21.Jun.18, 8:02:11	Alex			+1★	
21.Jun.18, 14:11:09	Maria		+1★		
21.Jun.18, 14:11:45	Maria			-2★	

Figure 6.4.: The visualization of $UI1_{TL+TD+GR}$ (transparency level = 5). The blank cell indicates that a group member has not adapted the rating of an item. Signs "+" or "-" represents that a group member has increased or decreased the rating of an item.



RATING ADAPTATION HISTORY OF GROUP MEMBERS

Timeline	Group ratings of items			Group recommendation
Starting time	2	3	3.25	
21.Jun.18, 8:02:11	2	3	3.5	
21.Jun.18, 14:11:09	2	3.25	3.5	
21.Jun.18, 14:11:45	2	3.25	3	

Figure 6.5.: The visualization of $UI2_{all}$ (transparency level = 5). Each number indicates the group rating of an item at a specific time. For instance, on 21.Jun.18, at 8:02:11, the group rating of the *"mountain"* option was 3.5 stars.

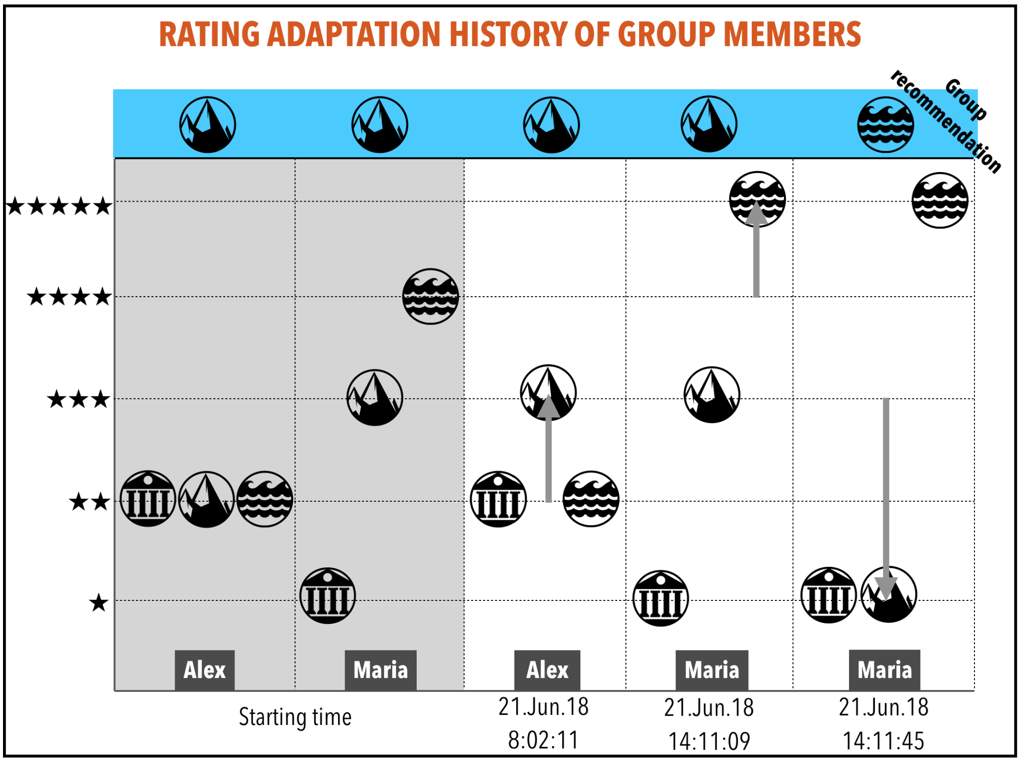


Figure 6.6.: The $UI1_{all}$ (transparency level = 6) visualizing all dimensions concerning the rating adaptations of group members.

highest transparency level could achieve the best performance in terms of decision manipulation counteraction. Besides, at a specific transparency level, we examine whether the ability of the UIs to counteract decision manipulation differs and discover the best UI. Moreover, we also want to investigate the most understandable UI from the users' point of view. Another focus of our study is to examine if the dimensions represented in the UIs have different impacts on the decision manipulation behavior of users. Particularly, we investigate which dimensions strongly change users' mind concerning decision manipulation. We assume *"group member"*, *"rating"*, and *"group recommendation"* dimensions could prevent users from decision manipulation since compared to other dimensions, they seem to be very helpful for predicting decision manipulation attempts of group members. Therefore, these dimensions should be included in the UIs for avoiding decision manipulation. All the mentioned aspects are captured in the following research questions:

- RQ_1: *"Which transparency level of the rating adaptation history best helps to counteract decision manipulation?"*

- RQ_2: *"At a specific transparency level, which UI performs the best in terms of decision manipulation counteraction?"*

- RQ_3: *"Which UI is the most understandable one?"*

- RQ_4: *"Which dimension in the rating adaptation history best helps to prevent users from decision manipulation?"*

Transparency level	Group	UI	Number of participants	
2	1	$UI1_{basis}$	120	
3	1	$UI1_{R}$	21	120
	1	$UI1_{TL}$	19	
	1	$UI1_{TD}$	21	
	1	$UI1_{GR}$	20	
	2	$UI2_{R}$	19	
	2	$UI2_{TD}$	20	
4	1	$UI1_{R+TD}$	26	120
	1	$UI1_{TL+TD}$	25	
	1	$UI1_{TL+GR}$	25	
	1	$UI1_{TD+GR}$	21	
	2	$UI2_{TD+GR}$	23	
5	1	$UI1_{R+TL+TD}$	32	120
	1	$UI1_{R+TD+GR}$	31	
	1	$UI1_{TL+TD+GR}$	28	
	2	$UI2_{all}$	29	
6	1	$UI1_{all}$	120	

Table 6.5.: The distribution of the participants for the UIs at different transparency levels.

6.5.2. User Study

To address the research questions, we conducted a user study with staff members and students from two universities[§]. In total, there were 120 participants (*males*: 45.83%, *females*: 54.17%) from 18 to 50 years old. Our user study was performed in the following steps:

Step 1 - Distribute UIs to participants: Each user study participant was provided with *a scenario description* (see *Section 6.4.2*) and *a sequence of five UIs* at five different transparency levels. To avoid possible biases, the UIs in each sequence were shown to the participant in random order. Moreover, the UIs were distributed to the participants using a *between-subjects* method in which each participant received a different UI sequence. At a certain time, the participant observed and evaluated *only one UI* in the provided sequence. Each UI was *indepedently* evaluated by the participant, i.e., *the evaluation of one UI did not rely upon the evaluations of other UIs*. Besides, the UIs were distributed in such a way that the total number of participants who observed the UIs at each transparency level should be equal. In this study, at each transparency level, there were in total 120 participants. Moreover, at a specific transparency level, we distributed the UIs so that the number of participants for each UI was balanced (see Table 6.5). However, the collected data shows that these numbers were slightly different. The reason was that some participants did not complete the user study and therefore some UIs were not evaluated by them.

Step 2 - Define criteria to evaluate the UIs: At a certain time, each participant observed one UI and evaluated its *understandability* using a 5-point Likert scale ranging from 1 (*completely not understandable*) to 5 (*completely understandable*). After that, the participant had to answer the following question: *"Assume, you were Maria in the mentioned scenario. If you had known that your rating adaptations would be shown to all group members as in the user interface, then what would you have done?"*.

Each participant had to answer this question by choosing one out of three values in the scale of [1..3]. This scale measures *the preparedness level* of the participant with regard to decision manipulation, where

[§]Hue University of Economics, Hue, Vietnam and Graz University of Technology, Graz, Austria

1: "manipulate", *2: "not sure/hesitate"*, and *3: "not manipulate"*. That means, *"the higher the value, the lower the preparedness level of the participant regarding decision manipulation"*. Also, the participant was asked to give a brief *explanation* of his/her answer. In case the participant decided *"not to manipulate"* the decision, he/she had to additionally specify the *influence level* of each dimension on his/her decision concerning decision manipulation. The *influence level* of a dimension was measured by a 5-point Likert scale ranging from 1 to 5, in which *"1"* indicates that the dimension *did not change* the participant's mind regarding decision manipulation and *"5"* indicates that the dimension made the participant *give up* decision manipulation.

6.6. Data Analysis and Results

6.6.1. Data Analysis

To address the research questions, we collected the evaluations of the participants for the UIs as follows:

- RQ_1: We separately collected the participants' answers for the UIs of *Group 1* and *Group 2*. In each group, at a specific transparency level, we gathered the participants' answers for all UIs.
- RQ_2: For each transparency level, we collected the participants' answers for all UIs from both *Group 1* and *Group 2*. In total, we had five sets of answers corresponding to five transparency levels.
- RQ_3: We gathered the participants' evaluations regarding the *understandability* of all UIs from two groups of UIs.
- RQ_4: We filtered out the participants who decided *"not to manipulate"* the decision. After that, in each UI, we gathered the participants' evaluations concerning the *influence level* of the dimensions.

The collected evaluations share the same characteristics: (i) *independent* (i.e., the evaluation of a UI was independent to the evaluations of other UIs), (ii) *ordinal* (in the range of [1..3] or [1..5]), and (iii) *not normally distributed* (Shapiro-Wilk tests, significance level $\alpha = .05, p < \alpha$). Because of that, we used non-parametric tests (Kruskal-Wallis, $\alpha = .05$) to analyze these evaluations. Besides, for RQ_1 and RQ_2, we additionally ran follow-up Mann-Whitney U tests ($\alpha = .05$) on the same sets of evaluations to further consider the participants' evaluations between pairs of different UIs. Running many Mann-Whitney U tests in the same sets could trigger *Type I errors*[¶]. To control this, we applied a *Bonferroni adjustment* (Pallant, 2007) to adapt the significance level. The *revised significance level* of each test was $\alpha' = \frac{\alpha}{N}$, N is the number of tests.

6.6.2. Results

a. Research Question RQ_1

(RQ_1: *"Which transparency level of a rating adaptation history best helps to counteract decision manipulation?"*)

In *Group 2*, we found out that there were no statistically significant differences in the participants' preparedness levels of decision manipulation across different transparency levels (Kruskal-Wallis, $p = .564 > \alpha$). This means, in *Group 2*, it was unclear which transparency level best helps to counteract decision manipulation.

[¶] In hypothesis testing, a *Type I error* involves rejecting the null hypothesis (i.e., *"there are no differences among evaluation sets"*) when it is actually true (Pallant, 2007).

Transparency level	Mean rank
2	211.66
3	239.46
4	256.04
5	278.80
6	**288.10**

Table 6.6.: Kruskal-Wallis test ($\alpha = .05, p = .000$) across different transparency levels of the UIs in *Group 1*. The higher the mean rank, the lower the preparedness level of decision manipulation.

In *Group 1*, there were statistically significant differences in the participants' preparedness levels regarding decision manipulation across different transparency levels (Kruskal-Wallis, $p = .000 < \alpha$). The mean ranks in Table 6.6 shows that at the *transparency level of 6*, the $UI1_{all}$ achieved the lowest preparedness levels, whereas the $UI1_{basic}$ (*transparency level = 2*) reported the highest. Besides, by performing ten follow-up Mann-Whitney U tests ($\alpha' = \alpha/10 = .005$) between pairs of *five* transparency levels, we found out that there was a statistically significant difference in the participants' preparedness levels between the UIs at the transparency levels of 2 and 6 ($p = .000 < \alpha'$).

The results can be explained as follows: At the transparency level of 2, the $UI1_{basic}$ only reveals the information of *"who has adapted the ratings of which items"*, which is too abstract and hard to detect decision manipulation attempts of group members. Therefore, this UI increased the participants' preparedness levels of decision manipulation. In contrast, at the transparency level of 6, the $UI1_{all}$ makes the rating adaptations of group members *completely transparent* (see Figure 6.6). Therefore, this UI effectively helped to discourage the participants from decision manipulation. Indeed, *80/120* participants who observed this UI decided *not to manipulate* the decision. *73.8%* of these participants mentioned that their rating adaptations were *too obvious*, and therefore others might recognize them as manipulators. The rest emphasized that decision manipulation should be avoided in any circumstance since this action is definitely a *'cheat'* and results in damaging the coherence among group members. Moreover, it is *"unfair"* if a group member tries to push the group recommendation to his/her favorite. They would feel *"so ashamed"* of themselves for doing such a behavior.

b. Research Question RQ_2

(RQ_2: *"At a specific transparency level, which UI performs the best in terms of decision manipulation counteraction?"*)

- *At the transparency level of 3*: The Kruskal-Wallis test reveals a statistically significant difference in the participants' preparedness levels regarding decision manipulation across different UIs ($p = .009 < \alpha$). An inspection of the mean ranks suggests that the $UI1_{GR}$ (depicted in Figure 6.2) achieved the lowest preparedness levels concerning decision manipulation (see Table 6.7). This means the $UI1_{GR}$ best helped to counteract decision manipulation. Differ from other UIs at the same transparency level, the $UI1_{GR}$ with the *"Group recommendation"* dimension helped the participants make sure that the second rating adaptation of Maria has pushed the group recommendation to the *"sea"* option. This could explain as to why *85%* of the participants were hesitant or gave up decision manipulation when observing the $UI1_{GR}$.
- *At the transparency level of 4*: The Kruskal-Wallis test reveals a significant difference in the participants' preparedness levels regarding decision manipulation across different UIs ($p = .018 < \alpha$). An inspection of the mean ranks suggests that the $UI1_{R+TD}$ best helped to counteract decision manipulation (see Table 6.8). Compared to other UIs at the same transparency level, only the $UI1_{R+TD}$

UI	Mean rank
$UI1_R$	61.69
$UI1_{TL}$	61.66
$UI1_{TD}$	64.93
$UI1_{GR}$	**79.45**
$UI2_R$	41.29
$UI2_{TD}$	52.80

Table 6.7.: Kruskal-Wallis test for all UIs at the *transparency level of 3* ($\alpha = .05, p = .009$). The higher the mean rank, the lower the preparedness level of decision manipulation.

UI	Mean rank
$UI1_{R+TD}$	**75.29**
$UI1_{TL+TD}$	62.76
$UI1_{TL+GR}$	53.68
$UI1_{TD+GR}$	63.57
$UI2_{TD+GR}$	45.93

Table 6.8.: Kruskal-Wallis test for all UIs at the *transparency level of 4* ($\alpha = .05, p = .018$). The higher the mean rank, the lower the preparedness level of decision manipulation.

(see Figure 6.3) explicitly shows the ratings (*original* and *adapted ratings*) of items. Moreover, each rating adaptation is additionally represented by a directed arrow, which clearly shows how group members have adapted the rating of the item. This could explain as to why this UI better helped the participants detect who might be a manipulator and kept them out of decision manipulation. 76% of the participants who observed the $UI1_{R+TD}$ were hesitant or decided not to manipulate the decision. They mentioned that they were afraid of being recognized as manipulators by others.

- *At the transparency level of 5*: The Kruskal-Wallis test reveals statistically significant differences in the participants' preparedness levels regarding decision manipulation across different UIs ($p = .038 < \alpha$). An inspection of the mean ranks suggests that the $UI1_{R+TL+TD}$ of *Group 1* seems to be the best UI in term of decision manipulation counteraction (see Table 6.9). However, we recognized that both $UI1_{R+TD+GR}$ and $UI1_{TL+TD+GR}$ also achieved similar preparedness levels (their mean ranks are quite close to the mean rank of the $UI1_{R+TL+TD}$). By running *three* follow-up Mann-Whitney U tests between pairs of these three UIs ($\alpha' = .05/3 = .017$), we found out that there were no statistically significant differences among them. That means, at the transparency level of 5, the UIs of *Group 1* have the same ability to counteract decision manipulation. These UIs effectively helped to discourage the participants from decision manipulation. More than 60% of the participants who observed these UIs avoided manipulating the decision. This could be explained by the fact that each UI represents five dimensions that reveal almost all information regarding the rating adaptations of group members. This helped the participants easily detect decision manipulation attempts of group members.

In addition, when answering the research question RQ_2, we also recognized that at every transparency level, the UIs of *Group 1* perform better than those of *Group 2* in terms of counteracting decision manipulation. Indeed, the mean ranks generated in the mentioned Kruskal-Wallis tests obviously show that at each transparency level, the UIs of *Group 2* always achieved higher preparedness levels with regard to decision manipulation (see Tables 6.7, 6.8, and 6.9). Moreover, the Mann-Whitney U tests confirm significantly differences in the preparedness levels between the UIs of *Group 1* and the UIs of *Group 2*. In particular, at the transparency level of 3, we found a significant difference between $UI1_{GR}$ and $UI2_R$ (Mann-Whitney

UI	Mean rank
$UI1_{R+TL+TD}$	**68.63**
$UI1_{R+TD+GR}$	62.21
$UI1_{TL+TD+GR}$	63.54
$UI2_{all}$	46.78

Table 6.9.: Kruskal-Wallis test for all UIs at the *transparency level of 5* ($\alpha = .05, p = .038$). The higher the mean rank, the lower the preparedness level of decision manipulation.

U test, $\alpha' = .003, p = .000$, *rank value($UI1_{GR}$)=25.63*, *rank value($UI2_{R}$)=14.08*). At the transparency level of 4, a significant difference was found between $UI1_{R+TD}$ and $UI2_{TD+GR}$ (Mann-Whitney U test, $\alpha' = .005, p = .002$, *rank value($UI1_{R+TD}$)=30.29*, *rank value($UI2_{TD+GR}$)*=19.02). Finally, at the transparency level of 5, a significant difference was triggered between $UI1_{R+TL+TD}$ and $UI2_{all}$ (Mann-Whitney U test, $\alpha' = .0083, p = .006$, *rank value($UI1_{R+TL+TD}$)=36.25*, *rank value($UI2_{all}$)=25.21*). According to these tests, it is obvious that the UIs of *Group 2* always achieved *significantly higher preparedness levels* of decision manipulation compared to the UIs of *Group 1*. This could be explained by the fact that the UIs of *Group 2* do not include the *"Group member"* dimension (see the visualization of a UI of *Group 2* in Figure 6.5). One common explanation given by the participants who observed the UIs of *Group 2* was that: *"It was very hard to track who has adapted the ratings of items"*. Therefore, compared to the UIs of *Group 1*, those of *Group 2* increased the participants' preparedness levels concerning decision manipulation.

c. Research Question RQ_3

(RQ_3: *"Which UI is the most understandable one?"*)

The Kruskal-Wallis test reveals a statistically significant difference in the understandability levels across different UIs ($p = .008 < \alpha$). The mean ranks in Table 6.10 show that the $UI1_{TL+TD+GR}$ (see Figure 6.4) was the most understandable UI. 78% of the participants who observed this UI found it *understandable* or *compleletely understandable*. The average score for the understandability of this UI was 4.2/5. Some typical comments regarding the understandability of this UI are *"it is very intuitive and understandable"*, *"it shows how the ratings of items have been adapted'*, or *"it reveals possible attempts of decision manipulations'"*.

d. Research Question RQ_4

(RQ_4: *"Which dimension in the rating adaptation history best helps to prevent users from decision manipulation?"*)

The Kruskal-Wallis tests show that for the UIs at the transparency levels of 2, 3, and 4, there were no significant differences in the influence levels across different dimensions. In other words, the dimensions represented in these UIs have equal impacts on the participants' decision manipulation behaviors. However, two exceptions were found in $UI1_{TL+TD+GR}$ (see Figure 6.4) and $UI1_{all}$ (see Figure 6.6). In these UIs, the Kruskal-Wallis tests reveal statistically significant differences in the influence levels across different dimensions ($p(UI1_{TL+TD+GR}) = .024 < \alpha$ and $p(UI1_{all}) = .006 < \alpha$). For the $UI1_{TL+TD+GR}$, an inspection of the mean ranks suggests that the *"Group recommendation"* dimension had the strongest influence on the participants' mind concerning decision manipulation (see Table 6.11). For the $UI1_{all}$, the *"Rating"* dimension best helped to prevent the participants from decision manipulation (see Table 6.12). Besides, the mean ranks in Table 6.11 and Table 6.12 show that the *"Timeline"* dimension did not have so

Transparency level	Group	UI	Mean rank
2	1	$UI1_{basis}$	301.05
3	1	$UI1_R$	363.17
	1	$UI1_{TL}$	340.66
	1	$UI1_{TD}$	343.40
	1	$UI1_{GR}$	339.85
	2	$UI2_R$	294.53
	2	$UI2_{TD}$	309.50
4	1	$UI1_{R+TD}$	297.35
	1	$UI1_{TL+TD}$	242.68
	1	$UI1_{TL+GR}$	263.00
	1	$UI1_{TD+GR}$	342.14
	2	$UI2_{TD+GR}$	305.15
5	1	$UI1_{R+TL+TD}$	305.83
	1	$UI1_{R+TD+GR}$	206.19
	1	$UI1_{TL+TD+GR}$	**380.95**
	2	$UI2_{all}$	325.28
6	1	$UI1_{all}$	273.34

Table 6.10.: Kruskal-Wallis test ($\alpha = .05, p = .008$) in the understandability levels across different UIs.

Dimension	Mean rank
GM	44.53
I	45.72
TL	31.06
TD	46.78
GR	**59.42**

Table 6.11.: Kruskal-Wallis test in the influence levels across different dimensions of the $UI1_{TL+TD+GR}$ ($\alpha = .05, p = .024$).

Dimension	Mean rank
GM	262.60
I	242.69
R	**272.73**
TL	198.57
TD	225.40
GR	258.99

Table 6.12.: Kruskal-Wallis test in the influence levels across different dimensions of the $UI1_{all}$ ($\alpha = .05, p = .006$).

much influence on changing the participants' mind with regard to decision manipulation. In other words, in the $UI1_{TL+TD+GR}$ and $UI1_{all}$, the *"Timeline"* did not help to avoid decision manipulation.

6.7. Conclusion and Future Work

In this chapter, we proposed different UIs and investigated which of them are the most understandable and most effective for counteracting decision manipulation in group recommender systems. The outcomes of our study further confirm the *Hawthorne effect* in the context of decision manipulation, which means if group members know that others could see their rating adaptations, then they tend to avoid decision manipulation. This work also provides practitioners with a hint to design UIs which help to avoid decision manipulation issues in group recommender systems.

To the best of our knowledge, up to now, there does not exist any research which proposes UI-driven solutions to counteract decision manipulation in group recommender systems. Therefore, in our work, we faced the difficulty in specifying *a baseline* to evaluate our proposed UIs. Another limitation of our research lies in the decision manipulation context. In this work, we discussed a manipulation issue in group decisions where users *know each other* and make decisions on *small sets of items*. Within the scope of future work, we will expand our study by proposing UIs which counteract decision manipulation in other contexts (e.g., group members have no relationship and decide on a large set of items). Alternatively, the UIs could also be evaluated in *other decision manipulation scenarios*. For instance, a group member alters item ratings to make the favorite options of his/her opponents never be chosen by the system. Besides, the evaluation process will be done with *a bigger set of observations* (in our study, the current sample data set of 120 observations for 17 UIs is quite limited) and with an engagement of *group dynamics aspects* (i.e., *age, gender, education background*, *personality of group members, etc.*).

Chapter 7

Recommender Systems in the Healthy Food Domain

This chapter is based on the results documented in (Tran et al., 2018a). The author of this thesis provided major distributions in terms of literature research and writing the whole chapter.

7.1. Abstract

Recently, food recommender systems have received increasing attention due to their relevance to healthy living. Most existing studies on the food domain focus on recommendations that suggest proper food items for individual users based on his/her preferences or health problems. These systems also provide functionalities to keep track of nutritional consumption as well as to persuade users to change their eating behavior in positive ways. Also, group recommendation functionalities are very useful in the food domain, especially for some scenarios when a group of users wants to have dinner together at home or to have a birthday party in a restaurant. Such scenarios create many challenges for food recommender systems since the preferences of all group members have to be taken into account adequately. In this chapter, we present an overview of recommendation techniques for individuals and groups in the healthy food domain. Besides, we analyze the existing state-of-the-art in food recommender systems and discuss research challenges related to the development of future food recommendation technologies.

7.2. Introduction

According to the prediction of the World Health Organization*, the quantity of overweight adults all over the world has reached an alarming number with 2.3 billions by 2015. More significantly, overweight and obesity also cause many chronic diseases (Robertson, 2004). An appropriate dietary intake is considered as an essential factor for improving overall well-being. Although most people are aware of the importance of healthy eating habits, they usually tend to neglect appropriate behaviors because of busy lifestyles and/or unwillingness to spend cognitive effort on food preparation. Those problems prevent users from healthy food consumption (Van Pinxteren et al., 2011). Hence, recommender systems are investigated as an effective solution in order to help users to change their eating behavior and to aim for healthier food choices.

However, food and diet are complex domains bringing many challenges for recommendation technologies. To generate recommendations, thousands of food items/ingredients have to be collected. Besides, since

*http://www.who.int

ingredients are usually combined in a recipe instead of being consumed separately, this exponentially increases the complexity of a recommender system (Freyne and Berkovsky, 2010). Moreover, food recommender systems not only recommend food suiting users' preferences but also suggest healthy food choices, keep track of eating behaviors, and convince users to change eating behaviors.

While many existing recommender systems mainly target individuals, there is a remarkable increase of recommender systems which generate suggestions for groups. Some early systems were developed in a variety of domains, such as, *group web page* recommendation (Lieberman et al., 1999a), *tour packages* for groups of tourists (Ardissono et al., 2003), *music tracks* and *playlists* for large groups of many listeners (Crossen et al., 2002), *movies* and *TV programs* for friends and family (O'Connor et al., 2001; Yu et al., 2006). Group scenarios are especially popular in the *food domain* in which a group of family members, friends, or colleagues wants to make a party or simply have a meal together. However, the complexity significantly increases when food recommender systems need to take into account the preferences of all group members as well as strategies for achieving the consensus within group members.

In this chapter, we summarize existing research related to food/recipe recommender systems which give recommendations based on the users' preferences and their nutritional needs. In this context, we also discuss scenarios for applying group recommender systems in the healthy food domain. An overview of some research related to the application of recommender systems in the healthy food domain is provided in Table 7.1.

The contributions of this chapter are the following. *First*, we provide a short overview of recommendation approaches for individuals. *Second*, we discuss *group decision making* issues which have an impact on the development of group recommendation technologies. *Third*, on the basis of categorizing food recommender systems, we analyze how well those systems can help individuals or groups to choose healthy food, which best fits their preferences and health situations. *Finally*, we point out some challenges of food recommender systems with regard to *user information*, *recommendation algorithms*, and *group decision making* as topics for future work.

The remainder of the chapter is organized as follows. In *Section 7.3*, we provide an overview of basic recommendation techniques for individuals and groups. In *Section 7.4*, we summarize existing studies on food recommender systems for single users and categorize them according to different criteria, such as *preferences, nutritional needs, health problems,* and *eating behaviors of users*. Besides, in this section, we also discuss some research related to food recommender systems in group scenarios. Research challenges for food recommender systems are discussed in *Section 7.5*. The chapter is concluded with *Section 7.6*.

7.3. Recommender Systems

Due to heavy information overloads triggered by the Internet, extracting/finding valuable information becomes increasingly difficult. In this context, recommender systems became an effective tool to extract useful information and deliver it efficiently. A recommender system predicts the preferences of users for unrated items and recommends new items to users. Along with the benefits of recommender systems, developing new recommendation approaches and including them in different fields rise immensely. The following subsections present an overview of recommendation techniques for individuals and groups.

7.3.1. Recommendation Techniques for Individuals

According to (Burke et al., 2011) and (Burke, 2000), a recommender system can be defined as follows: *"Any system that guides a user in a personalized way to interesting or useful objects in a large space of*

Food Recommender Systems	**Papers**	**Recommendation approaches**	**Functionality**
Type 1: Considering user preferences	El-Dosuky et al. (2012)	*Knowledge-based recommendation*	Proposing a framework for a Personalized nutrition service with knowledge-based recommendation.
	Freyne and Berkovsky (2010)	*CB, CF, Hybrid recommendation*	Predicting item ratings by breaking down recipes into ingredients and vice versa.
	Freyne et al. (2011)	*CB, CF, Hybrid recommendation*	Improving the quality of recommendations by using machine learning techniques and an understanding of user reasoning.
	Svensson et al. (2000)	CF	Developing an on-line grocery store to provide users with recipe recommendations by analyzing the social navigation in groups.
	Elahi et al. (2015)	*Matrix factorization*	**ChefPad** - Generating food recommendations by eliciting users' long-term and short-term preferences.
	Kuo et al. (2012)	*Graph-based recommendation*	Recommending sets of recipes by using user-specified ingredients.
Type 2: Considering nutritional needs of users	Ueta et al. (2011)	*Goal-oriented recipe recommendation* which suggests the right type of nutrient to treat users' health problems	Helping users to deal with health problems (e.g., acnes)
	Aberg (2006)	*Hybrid recommendation (CB & CF), Constraint-based recommendation*	**Meal Planning System** - Aiding the elderly to deal with malnutrition problems and change the food consumption behavior.
Type 3: Balancing between user preferences and nutritional needs of users	Elsweiler et al. (2015)	*CB*	Applying different approaches to bring healthiness aspects into recommender systems.
Type 4: Food recommender systems for groups	Berkovsky and Freyne (2010)	*CF, group-based recommendation*	Applying different aggregation strategies and user weighting models to generate recipe recommendations to a group of users.
	Elahi et al. (2014)	*Group recommendation, Critiquing-based conversational recommendation*	**ChefPad** - Generating food recommendations to groups by exploiting users' tags and ratings.

Table 7.1.: A summary of state-of-the-art of recommender systems in the healthy food domain (CF: Collaborative filtering recommender systems, CB: Content-based recommender systems)

possible options or that produces such objects as output". Recommender systems are intensively applied to recommend products and services (e.g., *movies, books, digital cameras*, and *financial services*) which best meet users' needs and preferences. Recently, in the healthy food domain, recommender systems have been discovered as a potential solution to help users to cope with the vast amount of available data related to foods/recipes. Many different techniques have been proposed for making personalized recommendations and these will be discussed in the followings.

Collaborative filtering recommender systems (CF). CF became one of the most researched techniques of recommender systems. The basic idea of CF is to use the wisdom of the crowd to make recommendations. First of all, a user implicitly or explicitly rates some given items. Then, the recommender identifies the *nearest neighbors* whose tastes are similar to those of a given user and recommends items that the nearest neighbors have liked (Ekstrand et al., 2011). CF is usually implemented on the basis of the following approaches: *user-based* (Asanov, 2011), *item-based* (Sarwar et al., 2001), *model-based approaches* (Koren et al., 2009), and *matrix factorization* (Bokde et al., 2015).

Content-based recommender systems (CB). These systems can make a personalized recommendation by exploiting information of available item descriptions (e.g., genre and director of movies) and user profiles describing what the users like. The main task of a CB system is to analyze the information regarding user preferences and item descriptions consumed by the user, and then recommend items based on this information. Research in this area primarily focused on recommending items with textual content, such as *web-pages* (Pazzani et al., 1996), *books* (Mooney and Roy, 2000), and *documents* (Lang, 1995). There are different approaches applied to make recommendations to users, such as *Information Retrieval* (Balabanović and Shoham, 1997) or *Machine Learning algorithms* (Mooney and Roy, 2000).

Knowledge-based recommender systems (KBS). KBS are recognized as a solution for tackling some problems generated by classical approaches (e.g., *ramp-up problems* (Burke, 2000)). Moreover, these systems are especially useful in domains where the number of available item ratings is deficient (e.g., apartments, financial services) or when users want to define their requirements explicitly (e.g., *"the color of the car should be white"*). There are two main approaches for developing knowledge-based recommender systems: *case-based recommendation* (Bridge et al., 2005) and *constraint-based recommendation* (Felfernig and Burke, 2008). In addition, *critiquing-based recommendation* is considered as a variant of case-based recommendation. This approach uses users' preferences to recommend specific items, and then elicits users' feedback in the form of critiques to improve the recommendation accuracy (Burke, 2000). There are four basic steps in a knowledge-based recommendation setting:

- *Requirement specification*: Users can interact with a recommender system to specify their requirements.
- *Repair of inconsistent requirements*: If the recommender can not find a solution, it suggests a set of repair actions, i.e., it proposes alternatives to user requirements ensuring the identification of a recommendation (Felfernig et al., 2011).
- *Presentation of results*: A set of alternatives is delivered to the user. These are usually presented as a ranked list according to the item utility for the user (Felfernig et al., 2006).
- *Explanation*: For each presented alternative, the user can activate a corresponding explanation to understand as to why a specific item has been recommended (Felfernig et al., 2006).

Hybrid recommender systems (HRS). HRS are based on a combination of the above-mentioned techniques. According to (Ricci et al., 2010): *"A hybrid system combining techniques A and B tries to use the advantages of A to fix the disadvantages of B"*. For instance, CF methods have to face the *new-item* problem, whereas CB approaches can tackle this problem because the prediction for new items is usually based on the available descriptions of these items. Burke (2002) presented some hybrid approaches which combine both CF and CB, including *weighted*, *switching*, *mixed*, *feature combination*, *cascade*, *feature augmentation*, and *meta-level*.

7.3.2. Recommendation Techniques for Groups

Research on recommender systems as discussed in *Section 7.3.1* only focuses on recommending items to individual users. However, in reality, there is a high probability of situations where recommender systems should support a group of users. For instance, recommending a tourist package for a group of friends or choosing a Christmas party destination for the colleagues in a company. In such situations, *Group Recommender Systems* (Masthoff, 2011) are considered as an optimal solution. In this subsection, we present an overview of some basic aspects of group-based recommendation.

Aggregation strategies. The main problem that group recommender systems need to solve is how to aggregate preferences based on information about the interests of individual group members. Masthoff (2011) presented many different strategies to merge individual user profiles into a group profile. These strategies can also be used for combining individual recommendations into group recommendations. Mostly used aggregation strategies for group recommendations are *least misery* (O'Connor et al., 2001), *average* (Ardissono et al., 2003), and *multiplicative* (Masthoff, 2004).

Group formation. In group recommendation scenarios, group creation and group maintenance are important steps that should be addressed. Groups can be built intentionally by explicit definition from the users (Smith et al., 1998) or unintentionally by an automatic identification from the system (McCarthy and Anagnost, 1998). Within a group, the roles of group members can be conferred differently according to their importance level within the group (Cantador and Castells, 2012; Berkovsky and Freyne, 2010). For instance, in a holiday planning scenario of a family, parents have more influence on choosing a tourism destination than children.

Group recommendation approaches. Group recommendations are mostly determined by using an *aggregated model* or an *aggregated prediction* (Jameson and Smyth, 2007).

- *Aggregated model* generates predictions for a group on the basis of aggregating individual user preferences into a *group profile*. The group recommendation process can be executed in three steps: *First*, users with similar preferences will be classified in subgroups. *Next*, the available items will be ranked based on each subgroup's preference. *Finally*, related items in subgroups are merged to get the ranking for the whole group. This approach was applied in some well-known systems, e.g., MUSICFX (McCarthy and Anagnost, 1998) and INTRIGUE (Ardissono et al., 2003) for the purpose of supporting groups of users to choose suitable alternatives.

- *Aggregated prediction* firstly computes the recommendation *for each group member* and then computes the intersection of individual recommendations to get the common recommendations for whole the group. For instance, POLYLENS (O'Connor et al., 2001) generates a ranked list of movies for each group member by using a classic CF approach. After that, the individual ranked lists are merged according to the *least misery* strategy, i.e., the group's happiness is the minimum of the individual members' happiness scores.

Group decision making. After forming groups, discovering some constraints within a group is an important phase which enables a recommender to make group recommendations. For instance, in the scenario of recommending recipes to a group of family members, because of the *seafood allergy* of one family member, recipes including *shrimp* or *sea-crab* might not be recommended to the whole group.

Besides, in group recommender systems, sometimes, knowing the preferences of other group members will have an impact on the decisions of other users. TRAVEL DECISION FORUM (Jameson et al., 2004) provides an interactive environment that allows members to optionally view (or copy) the preferences already specified by other group members. The preference visibility helps users save time and minimize conflicts generated in the decision making process (Jameson et al., 2004). However, in some decision scenarios, the insight into individual preferences of all group members can deteriorate the quality of the decision outcome (Stettinger et al., 2015). This issue was known as an *anchoring effect* ((Adomavicius et al., 2011; Felfernig, 2014), which is responsible for decisions biased by a shown reference value. In the context of group decision scenarios, the anchoring effect can be controlled by not completely disclosing the preferences of other group members in the early stages of the decision process (Felfernig et al., 2012b). In CHOICLA (Stettinger et al., 2015), a user can solely see the summary of all given ratings of other group members for a specific alternative after giving his/her rating. Seeing the summarized rating prevents all users from statistical inferences, which can influence on the quality of decision processes.

Until now, group recommendations are still a novel area compared to research on individual recommendations (Masthoff, 2011). There are still open issues on group decision making which need to be resolved in the future research, such as *bundle recommendations, intelligent user interface design, group aggregation strategies for cold-start problems* (Masthoff, 2011), *consensus achievement within group members, and counteracting decision biases* in group decision processes (Felfernig et al., 2014a).

7.4. Food Recommender Systems

"Where should we go for lunch?" or *"What should we eat for dinner?"* are the usual questions we have to answer very frequently. While many recommender systems only tried to match users' preferences to music, movie, or book domains, recently they also have been applied in the food domain in order to give reliable answers to the above questions. For instance, *RecipeKey*[†] is a food recommender system that filters recipes on the basis of considering favorite ingredients, existing food allergies, and item descriptions (e.g., meal-type, cuisine, preparation time, etc.) chosen by users.

Concerning food consumption these days, it is noticeable that there has been an increase in lifestyle-related illnesses, such as diabetes and obesity, which are the cause of many chronic diseases (Robertson, 2004). This problem can be improved by applying appropriate dietary (Knowler et al., 2002). In this context, food recommender systems are also investigated as a potential means to aid people to nourish themselves more healthily (Elsweiler et al., 2015). It makes sense to utilize food recommender systems as a part of a strategy for changing the eating behavior of users. In this context, food recommender systems not only learn users' preferences for ingredients and food styles, but also select healthy food by taking into account the health problems, nutritional needs, and previous eating behaviors of users.

As mentioned in (Mika, 2011), there are two types of food recommender systems. The first type (*Type 1*) recommends *healthier recipes* or *food items* which are the most similar to the ones the user liked in the past. The second type of recommender system (*Type 2*) only recommends to users those items which *have been identified beforehand by health care providers*. Besides, in this section, we also discuss two other types of food recommender systems (*Type 3* and *Type 4*), which consider other scenarios when making recommendations. *Type 3* generates recommendations on the basis of *considering the above criteria* to balance between the food that users like and the food that users should consume. All these types of recommender systems are primarily designed for individual users. *Type 4* represents *group recommendations* in which food items are consumed by groups of users rather than by individuals. These four types of food

[†]http://www.recipekey.com

recommender systems will be made more explicit and discussed in more detail in the following subsections.

7.4.1. Type 1: Considering User Preferences

In the healthy food domain, learning user tastes is recognized as a crucial pre-requisite step to suggest dishes that users will like. All research discussed in this subsection aims to recommend food items or menus to individual users on the basis of exploring user tastes. Most of them use popular recommendation techniques (Svensson et al., 2000; Freyne and Berkovsky, 2010; El-Dosuky et al., 2012), and/or combine with different techniques to improve the quality of recommendation (Elahi et al., 2015; Kuo et al., 2012) (see Table 7.1).

First of all, we present a food recommender system proposed by El-Dosuky et al. (2012) with a simple scenario which only recommends individual food items to users. The authors used TF-IDF (Term Frequency-Inverse Document Frequency) term extraction method to create the user profile. Thereafter, they applied some computations to identifying the similarity between a recipe and the user profile. Besides, healthy and standard food databases, which have been extracted from the United States Department of Agriculture[‡] (USDA), were incorporated into the knowledge base. The knowledge base is a domain ontology consisting of classes, relationships, and instances of classes. In order to get a recommendation, each user manually rates the food items of a specific category (e.g., fruits, vegetables, meat, etc.) as relevant or non-relevant for his/her interest. After that, the recommender computes the similarity between the food items and the previously computed user profile. If the similarity value is higher than a predefined threshold, then the food item is recommended. Otherwise, it gets ignored.

In another research, Freyne and Berkovsky (2010) used a CB algorithm to predict the rating value for a target recipe on the basis of exploiting the information of corresponding ingredients included in this recipe. The prediction process includes the following steps:

- Break down an unrated target recipe r_t into ingredients $ingr_1$, $ingr_2$, ..., $ingr_n$.
- Assign the rating value for each ingredient in the target recipe r_t according to Equation 7.1 as shown below. Particularly, the rating value of the user u_a for a specific ingredient $ingr_i$ in the target recipe r_t (i.e., rat(u_a, $ingr_i$)) is calculated using the rating values of the user u_a for all other recipes r_l which contain the ingredient $ingr_i$ (i.e., rat(u_a, r_l)). The value l mentioned in Equation 7.1 is the number of recipes containing $ingr_i$.

$$rat(u_a, ingr_i) = \frac{\sum_{l\ s.t\ ingr_i \in r_l} rat(u_a, r_l)}{l} \tag{7.1}$$

- Predict the rating value of the user u_a for the target recipe r_t (i.e., pred(u_a, r_t)) based on the average of all the rating values of all ingredients $ingr_1$, ..., $ingr_j$ included in this recipe (see Equation 7.2).

$$pred(u_a, r_t) = \frac{\sum_{j \in r_t} rat(u_a, ingr_j)}{j} \tag{7.2}$$

Recipes with a high predicted rating value will be recommended to user u_a. An illustration of predicting a rating value for a target recipe is presented in the following example:

Let us assume that $recipe_1$ is a recipe which has not been rated by user u_a. It includes 3 ingredients, i.e., $ingr_1$, $ingr_2$, and $ingr_3$. $ingr_1$ is included in $recipe_4$ and $recipe_2$, $ingr_2$ is included in $recipe_3$, and $ingr_3$ is included in $recipe_2$ and $recipe_3$. Rating values of user u_a for $recipe_2$, $recipe_3$, and $recipe_4$ are respectively

[‡]https://ndb.nal.usda.gov/

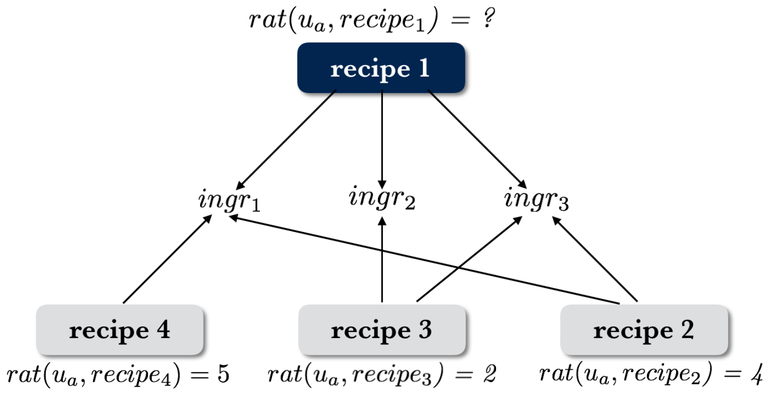


Figure 7.1.: Predicting the rating value for a target recipe using a CB algorithm proposed by Freyne and Berkovsky (2010)

4, 2, and 5 (see Figure 7.1).

According to Equation 7.1, rating values for the ingredients of $recipe_1$ will be evaluated as follows:

rat(u_a,$ingr_1$) = $\frac{rat(u_a,recipe_4)+rat(u_a,recipe_2)}{2}$ = $\frac{5+4}{2}$ = 4.5

rat(u_a,$ingr_2$) = $rat(u_a, recipe_3)$ = 2

rat(u_a,$ingr_3$) = $\frac{rat(u_a,recipe_2)+rat(u_a,recipe_3)}{2}$ = $\frac{4+2}{2}$ = 3

The prediction value of $recipe_1$ for user u_a is calculated by applying Equation 7.2 as follows:

pred(u_a,$recipe_1$) = $\frac{rat(u_a,ingr_1)+rat(u_a,ingr_2)+rat(u_a,ingr_3)}{3}$ = $\frac{4.5+2+3}{3}$ = 3.166

Recently, some new approaches have been included to food recommender systems, such as *using labels for different clusters of users* (Svensson et al., 2000), *active learning algorithms*, and *matrix factorization* (Elahi et al., 2015). Particularly, Svensson et al. (2000) designed an on-line food shop to suggest kinds of food that should be purchased by users. Based on recipes that users have chosen before, user groups are labeled and named according to their content, such as *"meat lovers"*, *"vegetarians"*, and *"spice lovers"*. The recommended recipes are determined based on three different characteristics chosen by users: *user groups*, *food categories* (e.g., fish, oriental, Italian, red meat, chicken), and *ingredients* (e.g., rice, spaghetti, curry, tomatoes). Users select recipes from the recommendation list and put them into a shopping basket. Then, all ingredients of the chosen recipes are automatically added to the list of items which is delivered to a user's doorstep. Besides, to enhance the social interaction for recipes, some additional features (e.g., the average rating value or comments from other users) are added to each recommended recipe.

Elahi et al. (2015) proposed a food recommender system by using an *active learning algorithm* and *matrix factorization*. This research provides users with a complete human-computer interaction to collect *long-term user preferences* in terms of recipe ratings and tags. In addition, when requesting recommendations, users are required to provide *short-term preferences* referring to ingredients which they want to cook or to include in the meal. Then, the system utilizes both types of user preferences to make recommendations. The long-term preferences are exploited by a *Matrix Factorization rating prediction*

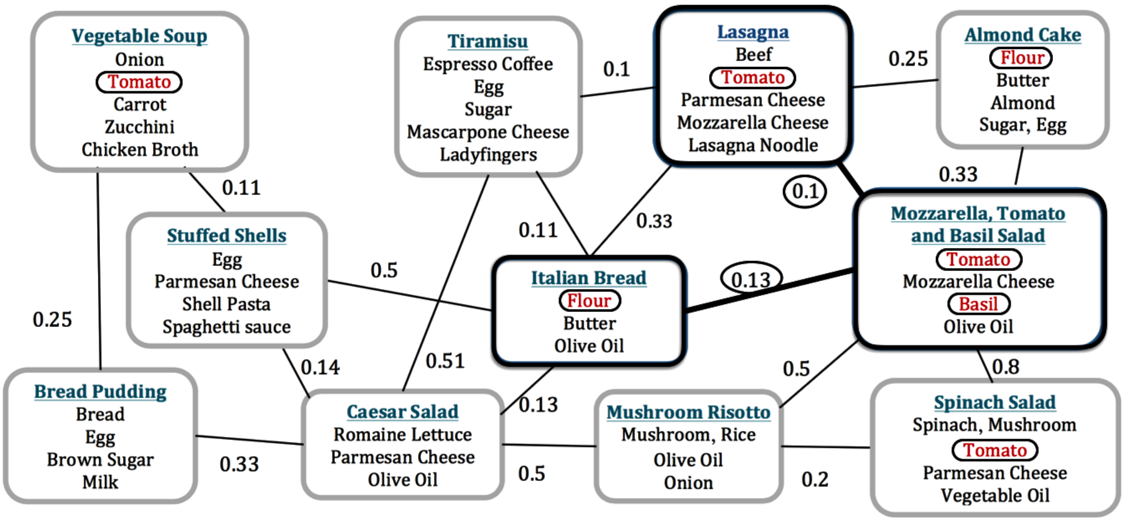


Figure 7.2.: An example of recipe graph *G* for menu planning (Kuo et al., 2012). *"Tomato, flour, basil"* (ingredients shown with black borders) are query ingredients. The recommended menu plan is a set of recipes {*"Mozzarella, Tomato, and Basil Salad", "Lasagna", "Italian Bread*"} (nodes shown with black frames) which the total menu cost is minimal (i.e., 0.23).

model which is designed to consider both user tags and ratings. Each user and each recipe are modeled by vectors that represent their latent features. The rating value of a user for a specific item is estimated by computing the inner product of the user and item vectors. With short-term preferences, the system filters recipes according to the current user preferences. The recipes with the highest rating values are recommended to the user.

While most of the existing research in the food domain primarily focuses on making recommendations on food items or recipes, there is a need for users to plan menus with the combination of many recipes into complete meals. With this idea, Kuo et al. (2012) proposed an intelligent menu planning mechanism that suggests a set of recipes using a *graph-based algorithm*. First, an undirected recipe graph is constructed, where *each node* is a recipe possessing a set of ingredients, *each edge* represents the relationship between two recipes, and *edge weight* represents the distance between two recipes (see Figure 7.2). The weight of each edge connecting two different recipes describes *the cost* of a menu consisting of these two recipes. The lower the weight, the higher the probability two recipes co-occur in a menu. For instance, in Figure 7.2, the recipe *"Italian Bread"* has a co-occurrence relationship with five recipes, i.e., *"Tiramisu"*, *"Lasagne"*, *"Mozzarella, Tomato, and Basil Salad"*, *"Caesar Salad"*, and *"Stuffed Shells"*. Among these five recipes, *"Tiramisu"* has the highest co-occurrence relationship with *"Italian Bread"* since the weight of their edge is lowest (i.e., 0.11). Whereas, *"Stuffed Shells"* has the lowest relationship with *"Italian Bread"* because the weight of their edge is highest (i.e., 0.5).

Besides, the menu cost is defined as the weighted sum of edges of the minimum spanning tree on the induced sub-graph. From that, a *menu plan* is created by choosing a set of recipes that contains all query ingredients (i.e., ingredients requested by users) and the menu cost is minimal. For instance, in Figure 7.2, with query ingredients {*tomato, flour, basil*}, we can find many different sets of recipes, for instance, {*"Mozzarella, Tomato, and Basil Salad", "Lasagna", "Italian Bread"*}, {*"Mozzarella, Tomato, and Basil", "Lasagne", "Almond cake"*}, {*"Mozzarella, Tomato, and Basil","Italian Bread", "Spinach*

Salad"}, etc. However, the first set {*"Mozzarella, Tomato, and Basil Salad", "Lasagna", "Italian Bread"*} will be recommended to users because its total menu cost is minimal (i.e., 0.23).

7.4.2. Type 2: Considering Nutritional Needs of Users

Nowadays, unhealthy eating habits and imbalanced nutrition increase the possibilities of people having obesity and other dietary-related conditions such as diabetes, hypertension, etc. As a treatment or preventive measure, nutritionists or dietitians usually recommend regular exercises and design individualized meal plans for their patients. Unfortunately, these nutrition experts are overloaded with too many patients to manually tailor an individualized meal plan for each user. That is where food recommender systems can be used as an intelligent nutrition consultation system. In this subsection, we provide a discussion of recommender systems that takes into account nutritional needs (see Table 7.1).

First, we discuss a simple recommendation scenario showing how menu items can be recommended to users on the basis of considering their *nutritional needs* as well as *health problems*. In this context, a user enters some personal information (e.g., *age, gender, occupation, physical activities, health problem*, etc.). This information is the basis for selecting food items that best fit the user's nutritional needs. The following example will be an illustration of this scenario.

In a menu recommender system, we assume that there are *five* menus with corresponding information, e.g, *ingredients, calories, fat* (see Table 7.2). A user u_a enters the following information: *Age*: 52, *Gender*: male, *Occupation*: office worker, *Physical activity*: walking (10 minutes/day), *Health problem*: cardiovascular. For recommending appropriate menus to user u_a, the following steps should be performed:

Menus	**Main ingredients**	**Calories (kcal)**	**Fat(%)**
$menu_1$	butter, chicken, potato, cucumber, cream, garlic, salt, pepper	2010	27
$menu_2$	pork, mushroom, broccoli, paprika, green onion, oil, salt, pepper	2200	30
$menu_3$	chicken, mushroom, salad, onion, olive oil, tomato, salt, pepper	1500	21
$menu_4$	beef, shrimp, tomato, garlic, egg, salt, pepper	2400	31
$menu_5$	pork, bean, tomato, pumpkin oil, salad, egg, salt, pepper	1700	25

Table 7.2.: A list of available menus with corresponding information

- Step 1: An energy table from DACH[§] (see Table 7.3) is used to estimate the number of calories (in *kcal*) which the user u_a should get daily. The number of calories intake per day for each person is estimated according to *age, gender* and *PAL* (Physical Activity Level) value. PAL value is categorized into 3 types:
  - PAL = 1.4: Is used for people who have exclusively sedentary lifestyles (such as *office workers, precision mechanics*) with very little or no strenuous leisure activity.
  - PAL = 1.6: Is used for people who have sedentary lifestyles, but additional energy is required for long-time walking and standing activities, such as *laboratory assistants, students, production line workers*.

[§] http://www.sge-ssn.ch.

– PAL = 1.8: Is used for people who have extensive lifestyles, for instance, *sellers, waiters, mechanics, artisans*.

In this example, user u_a is an office worker with very little physical activity (only 10 minutes/day for walking), which means his PAL value belongs to the first type. By looking up the information regarding *age*, *gender* and *physical activity* from Table 7.3, we can find the daily calories intake for u_a is 2200 kcal.

- Step 2: Filtering menus with the number of calories smaller or equal *2200 kcal/day.*
- Step 3: Ranking filtered menus in ascending order of fat (since u_a has heart disease, less fatty menus will be shown to him first).

Adults (years old)	**Reference values for energy intake in kcal/day**					
	PAL-value (1.4)		PAL-value (1.6)		PAL-value (1.8)	
	male	*female*	*male*	*female*	*male*	*female*
15 to 19	2600	2000	3000	2300	3400	2600
19 to 25	2400	1900	2800	2200	3100	2500
25 to 51	2300	1800	2700	2100	3000	2400
51 to 65	2200	1700	2500	2000	2800	2200
> 65	2100	1700	2500	1900	2800	2100

Table 7.3.: Reference values for nutritional intake. Bonn. 2. Edition, 1. Volume (2015) published by German Nutrition Association, Austrian Nutrition Association, and Swiss Nutrition Association.

In Table 7.4, we can see that $menu_4$ will not be added to the recommendation list because its calories are more than 2200 kcal. The list of recommended menus is ranked in ascending order of fat.

Menus	**Main ingredients**	**Calories (kcal)**	**Fat(%)**
$menu_3$	chicken, mushroom, salad, onion, olive oil, tomato, salt, pepper	1500	21
$menu_5$	pork, bean, tomato, pumpkin oil, salad, egg, olive oil, salt, pepper	1700	25
$menu_1$	butter, chicken, potato, cucumber, cream, garlic, salt, pepper	2010	27
$menu_2$	pork, mushroom, broccoli, paprika, green onion, oil, salt, pepper	2200	30

Table 7.4.: A list of recommended menus to user u_a

In order to improve the health conditions of users, Ueta et al. (2011) proposed a *goal-oriented recipe recommendation* to provide a list of dishes containing the right type of nutrient to treat users' health problems. To do that, first of all, a user enters her health problem in natural language, for instance, *"I want to cure my acnes"*. Next, the system analyzes the user's request and identifies the keywords describing the health issue (e.g., *acne*). The noun is pushed into the co-occurrence database to search the nutrient co-occurring mostly with it. For instance, by searching the noun *"acne"* in the co-occurrence database, *pantothenic acid* is found as a nutrient component which can be used for curing *acne* because it co-occurs with *"acne"* more often than any other nutrients. Finally, the nutrients identified in the previous step are used to find dishes that are closest to the nutrients in a food database. This food database includes two sub-databases: *ingredient nutrient database* and *nutritional information database* for recipes. The *ingredient nutrient database* contains information about the nutritional value of each ingredient. The *nutritional information database* includes recipe types and the number of nutrients contained in each recipe. The ingredients in each recipe are identified and their nutritional elements are calculated using the ingredient nutrient database. When recommending recipes for users, the system also considers the daily nutrient intake of users. These requirements vary according to *age* and *gender* of users.

In research related to dealing with malnutrition for the elderly, Aberg (Aberg, 2006) proposed a *menu-planning tool* which is required to take into account the following user-related information:

- Dietary restrictions, such as allergic ingredients;
- Nutritional values, such as the amount of fat or protein contained in a recipe;
- Preparation time of a meal;
- Preparation difficulty of a meal;
- Cost of necessary ingredients for a meal;
- The availability of ingredients for a meal;
- The variety of meals in terms of used ingredients and meal category;
- User food preferences, i.e., rating of a user for a certain recipe.

To be able to consider all these requirements, the author applied a hybrid design on the basis of combining *CF, CB*, and *constraint-based recommendation*. CF recommendation approach uses the ratings to predict the user's feedback on unrated recipes. CB recommendation approach uses *XML-based mark-up* language to represent the needed information for the recipes in the database. A constraint-based recommendation approach represented as a constraint satisfaction problem is used to construct optimal meal plans. A constraint satisfaction problem is modeled with two different approaches: *parameter-based* approach and *recipe-based* approach. However, the author did not mention in detail the *recipe-based approach*. Therefore, in this chapter, we solely discuss the *parameter-based approach* and the details of this approach are presented in Table 7.5. A prototype was developed to offer a meal-plan recommendation to users at a certain time. Users can switch between the top-5 meal plans and give ratings on the recommended recipes or create special settings for a meal.

For demonstration purposes, we propose an example of a constraint satisfaction problem, which is similar to a parameter-based approach (Aberg, 2006) to suggest a recipe on the basis of taking into account user's preferences. In this example, we assume that variables are used for representing the parameters of a recipe, such as *time, cost, energy, protein, allergies, disease*, where *time* (in *minutes*) is the preparation time of a recipe, *cost* (in *euro*) is the cost of a recipe, *energy* (in *kcal*) is the nutritional value of a recipe, *protein* (in %) is the percentage of protein contained in a recipe, *allergies* represents a set of allergic ingredients of users, and *disease* represents health problems of users. Each variable has a corresponding domain definition, for instance, dom(time) = [1..60]. Besides, a knowledge base *CKB* (Constraint Knowledge Base) includes constraints used for describing the knowledge base. For instance, $time < 60$ denotes the fact that *"preparation time of a recipe should be lower than 60 minutes"*. *PREF* is the set of user preferences, which should be consistent with *CKB* such that a corresponding solution can be identified.

- V = {*time, cost, ingredients, energy, protein, allergies, diseases*}
- D = {
  dom(*time*) = [1..60],

Constraint satisfaction problem	Parameter-based approach
Variables	time, cost, energy, protein
Variable domains	Variable domains are defined on the basis of existing values in the recipe database (e.g., *cost* = [1..100] represents the cost of a recipe can be from 0 to 100 Euros).
Knowledge base	* *Hard constraints:* - The constraint: $\{allergies = seafood \Rightarrow ingredients \neq sea-crab\}$ represents the knowledge that if a user is allergic to seafood then sea-crab should not be included into recommended recipes. * *Soft constraints*: - For the variety of recipes, recipes having many similar ingredients to the previous meals will not be chosen (e.g., *beef* and *potato* will not be chosen for dinner today because they were already consumed on lunch); - Recipes with high predicted rating will have higher probability to be recommended to users.

Table 7.5.: The constraint satisfaction problem modeled with a *parameter-based approach* (Aberg, 2006).

dom(cost) = [1..100],
dom(*energy*) = [1..3000],
dom(protein) = [1..100],
dom(*allergies*) = $[milk, egg, peanut, seafood, wheat]$,
dom(*diseases*) = $[diabetes, cardiovascular, parkinson, digestion, alzheimer, osteoarthritis, osteoporosis]$,
dom(*ingredients*) = $[vegetables, shrimp, sea-crab, fish, pork, beef, chicken, spices, butter, cheese, fruits]$
}

- CKB = {
$c_1 : time < 60$,
$c_2 : cost < 100$,
$c_3 : energy < 3000$,
$c_4 : protein < 35\%$,
$c_5 : disease = cardiovascular \Rightarrow protein < 30$,
$c_6 : allergies = seafood \Rightarrow ingredients \neq sea-crab$
}
- PREF = {
$pref_1 : time < 30$,
$pref_2 : cost < 50$,
$pref_3 : energy = 2200$,
$pref_4 : protein = 25\%$,
$pref_5 : allergies = seafood$,
$pref_6 : disease = cardiovascular$
}

On the basis of the constraint satisfaction problem as specified above, one solution can be determined for the user: $\{time = 25, cost = 40, ingredients = \{vegetables, chicken, spices, fruits\}, energy = 2200, protein = 25\%\}$.

7.4.3. Type 3: Balancing between User Preferences and Nutritional Needs of Users

Considering either user preferences or nutritional needs in an isolated fashion sometimes leads to sub-optimal recommendations of food items. For instance, if recommenders only take into account user preferences, then lousy eating habits would also be encouraged. On the contrary, if only nutritional needs are considered, then proposed food items sometimes will not be attractive to users. Therefore, considering both user preferences and nutritional needs seems to provide the best solution since when users receive more relevant recommendations, they become more interested and increasingly engaged in using them.

We now discuss a simple recommendation scenario showing how a food recommender system can suggest menu items on the basis of considering both user preferences and nutritional needs. In this example, we assume the existence of a menu table as shown in Table 7.2. A user u_a provides his personal information as follows: *Age*: 52, *Gender*: male, *Occupation:* office worker, *Physical activity*: walking (10 minutes/day), *Health problem*: cardiovascular, *Favorite ingredients*: tomato. In this scenario, the recommender system considers both ingredients preferred by user u_a and further user-related information (e.g., *age, gender, occupation, physical activity,* and *health problem*). The list of recommended menus is created by performing the following steps:

- *Step 1*: Estimating the daily number of calories for user u_a by looking up the energy table shown in Table 7.3. The user u_a is an office worker and has very little physical activity per day (only 10 minutes/day for walking), hence the nutrient intake of the user u_a is 2200 kcal.
- *Step 2*: Filtering out menus from Table 7.2 which contain lower or equal 2200 kcal of calories and include favourite ingredient *"tomato"*.
- *Step 3*: Ranking the filtered menus in ascending order of *"Fat"* (because u_a has vascular disease, less fatty menus will be shown to him first).

After accomplishing these steps, there are two menus (i.e., $menu_3$ and $menu_5$) will be recommended to user u_a (see Table 7.6).

Menus	**Main ingredients**	**Calories (kcal)**	**Fat(%)**
$menu_3$	chicken, mushroom, salad, onion, olive oil, tomato, salt, pepper	1500	21
$menu_5$	pork, bean, tomato, pumpkin oil, salad, egg, salt, pepper	1700	25

Table 7.6.: A list of menus recommended to user u_a on the basis of considering his favorite ingredients (i.e., *tomato*) and nutritional needs.

Also, to balance users' preferences and nutritional needs, Elsweiler et al. (2015) proposed two approaches to integrate nutritional aspects into recommendations.

- *The first approach* figures out trade-offs between giving the user some foods she really likes and some foods which are healthy for her. A food recommendation is generated in the following steps. First, a prediction algorithm estimates the top recipes for the user (i.e., a set of recipes with a predicted probability above a certain threshold). Next, the number of calories and the amount of fat (per gram) for each recipe in the chosen set are calculated. Finally, meals with less fat or calories (per gram) will be chosen as the final recommendation.
- *In the second approach*, instead of recommending individual meals, this approach proposes complete meal plans, which are generated not only based on the users' food preferences but also conform to

daily nutritional guidelines (Harvey and Elsweiler, 2015). For making recommendations, the user provides information regarding his/her preferences by rating several recipes in the system using a 5-star rating scale. Besides, the recommender also takes into account users' personal information, such as *height, weight, age, daily activity level*, and *goal (lose, gain, or maintain weight)* in order to calculate the nutritional needs. The nutritional requirements of users are calculated using an updated version of the *Harris-Benedict* equation (Roza and Shizgal, 1984). After that, the recommender predicts the ratings for unrated recipes and sends a ranked list of recipes with high ratings (e.g., 4 or 5 stars) to the *"Planner"*. The *"Planner"* takes *top-n recipes* from the ranked list of recipes and splits them into two separated sets: one for breakfasts and one for main meals. A full search is performed to find all combinations of these recipes in the sequence {Breakfast, Main meal, Main meal}, which meet the target nutritional needs. For instance, {*Muesli Breakfast Muffins, Catalan Chickpeas, Chicken Cacciatore*} (Harvey and Elsweiler, 2015) represents a complete menu recommended to users, where *Muesli Breakfast Muffins* is for breakfast, *Catalan Chickpeas* for lunch, and *Chicken Cacciatore* for dinner. Combinations with the same recipes can not be repeated, for instance, $\{r_1, r_2, r_3\}$ and $\{r_1, r_3, r_2\}$ are considered as only one menu plan.

Although two of the above-proposed approaches support the trade-off between users' preferences and healthy foods, the suitability of combining separate ingredients into a complete meal should be considered in more detail to make an appealing meal plan (Elsweiler et al., 2015).

7.4.4. Type 4: Food Recommender Systems for Groups

As mentioned above, in many real-world scenarios, recipe and food consumption are good examples of a group activity, for instance, a birthday party with friends or daily meals with family members (Elahi et al., 2014). In these scenarios, recommendations should be tailored to the entire group to assure the maximum satisfaction of each member and the group as a whole.

CF is one of the most widely used recommendation techniques and also applied in many group recommender systems (McCarthy et al., 2006; O'Connor et al., 2001). In the food domain, Berkovsky and Freyne (2010) investigated the applicability of two CF recommendation strategies to discover which strategy is the most relevant when making CF recommendations for a group. The authors discussed two group-based recommendation strategies as the following:

- **Aggregated models strategy**. First, this strategy computes a rating $rat(f_a, r_i)$ for a family f_a and recipe r_i by aggregating the individual ratings $rat(u_x, r_i)$ of family members $u_x \in f_a$ who rated recipe r_i according to their relative weight $\omega(u_x, f_a)$ (see Equation 7.3). The authors add weights into the rating calculation process to allow some users in a family to have more influence on the group decision than others. For instance, parents have more influence on the group decision than children, and therefore weights assigned to parents are higher than the children's ones. The details of weighting models will be presented in the next paragraph.

$$rat(f_a, r_i) = \frac{\sum_{x \in f_a} \omega(u_x, f_a) rat(u_x, r_i)}{\sum_{x \in f_a} \omega(u_x, f_a)} \quad (7.3)$$

  After that, CF is applied to the family model. Particularly, a prediction $pred(f_a, r_i)$ for the whole family f_a and unrated recipe r_i is generated by computing similarity degree $sim(f_a, f_b)$ between the family f_a and all other families $f_b \in F$, and then aggregating all family's ratings $rat(f_b, r_i)$ for recipe r_i according to the similarity degree $sim(f_a, f_b)$ (see Equation 7.4).

$$pred(f_a, r_i) = \frac{\sum_{f_b \in F} sim(f_a, f_b) rat(f_b, r_i)}{\sum_{f_b \in F} sim(f_a, f_b)} \quad (7.4)$$

- **Aggregated predictions strategy**. First, this strategy generates individual predictions $pred(u_x, r_i)$ for user u_x and unrated recipe r_i by using the standard CF algorithm (see Equation 7.5). In this prediction, the degree of similarity $sim(u_x, u_y)$ between the target user u_x and all other users $u_y \in U$ is calculated according to Equation 7.6, where k is the number of items already rated by the user u_x and the user u_y (Freyne et al., 2011). Then, individual ratings $rat(u_y, r_i)$ of users who rated r_i are aggregated according to the similarity degree $sim(u_x, u_y)$.

$$pred(u_x, r_i) = \frac{\sum_{y \in U} sim(u_x, u_y) rat(u_y, r_i)}{\sum_{i \in U} sim(u_x, u_y)} \tag{7.5}$$

$$sim(u_x, u_y) = \frac{\sum_{i=1}^{k} (u_{x_i} - \overline{u_x})(u_{y_i} - \overline{u_y})}{\sqrt{\sum_{i=1}^{k} (u_{x_i} - \overline{u_x})^2} \sqrt{\sum_{i=1}^{k} (u_{y_i} - \overline{u_y})^2}} \tag{7.6}$$

After that, to generate the prediction $pred(f_a, r_i)$ for the whole family f_a and recipe r_i, individual predictions $pred(u_x, r_i)$ of family members $u_x \in f_a$ are aggregated according to their relative weight $\omega(u_x, f_a)$ (see Equation 7.7).

$$pred(f_a, r_i) = \frac{\sum_{x \in f_a} \omega(u_x, f_a) pred(u_x, r_i)}{\sum_{x \in f_a} \omega(u_x, f_a)} \tag{7.7}$$

Both *aggregated models strategy* and *aggregated predictions strategy* recommend a list of recipes to the whole family by considering the task of recommending *top-k* recipes, i.e., k recipes having the highest predicted ratings.

The evaluation results on MAE (Mean Absolute Error) show that the *aggregated models strategy* are usually predominant to the *aggregated predictions strategy* (Berkovsky and Freyne, 2010). This means individual models of users should be aggregated into a group model first and then using this model in the recommendation process.

- **Weighting models**. Inspired by allowing some users to have more influence than others, the authors proposed four different weighting models when aggregating the data of individual users. Two first models (called *uniform* model and *role-based* model) assign pre-defined weights for users. Particularly, the *uniform* model uses the same weight for all group members. The *role-based* model weights users according to their role. For instance, there are two roles specified in a family party: *organizer* and *family member*. The weight for the *organizer* will be *2* because she is responsible for organizing the party as well as preparing food. Whereas, the weights for *family-members* are 1 because they are likely less important people. Two other models (called *role-based* model and *family-log* model) weight users according to their interactions with the content. The *role-based* model weights users according to their activities across the entire community. The activity of a certain user is predicted based on the number of ratings that (s)he rated for items. The *family-log* model weights users according to their activities in relation to other family members.

With the idea of combining individual user preferences into a group profile using aggregation heuristics (Masthoff, 2011) (e.g., *Least Misery, Average, Most Pleasure, Group Distance, Ensemble, etc.*), we discuss in this subsection a simple group recommendation scenario in the food domain to show how a group recommendation can be created.

Supposing that in a recipe recommender system, we have a group with four users (e.g., $user_1..user_4$) who rated five recipes (e.g., $recipe_1..recipe_5$) using a 5-star rating scale. We use *Least Misery* strategy (Masthoff, 2011) to aggregate individual user preferences into a group profile as a whole. *Least Misery strategy* makes sense in recipe decision scenarios since it helps to minimize the misery within a group. This means recipes which are not liked or can not be consumed by at least one group member will not be recommended to the whole group. In our example, the *group rating value* for each recipe is the minimum of all ratings given by all group members (see Table 7.7). After that, the recipe having the highest group rating value will be recommended to the group (Cantador and Castells, 2012). In this example, $recipe_1$ is recommended to the group since its group rating value is highest (i.e., 4).

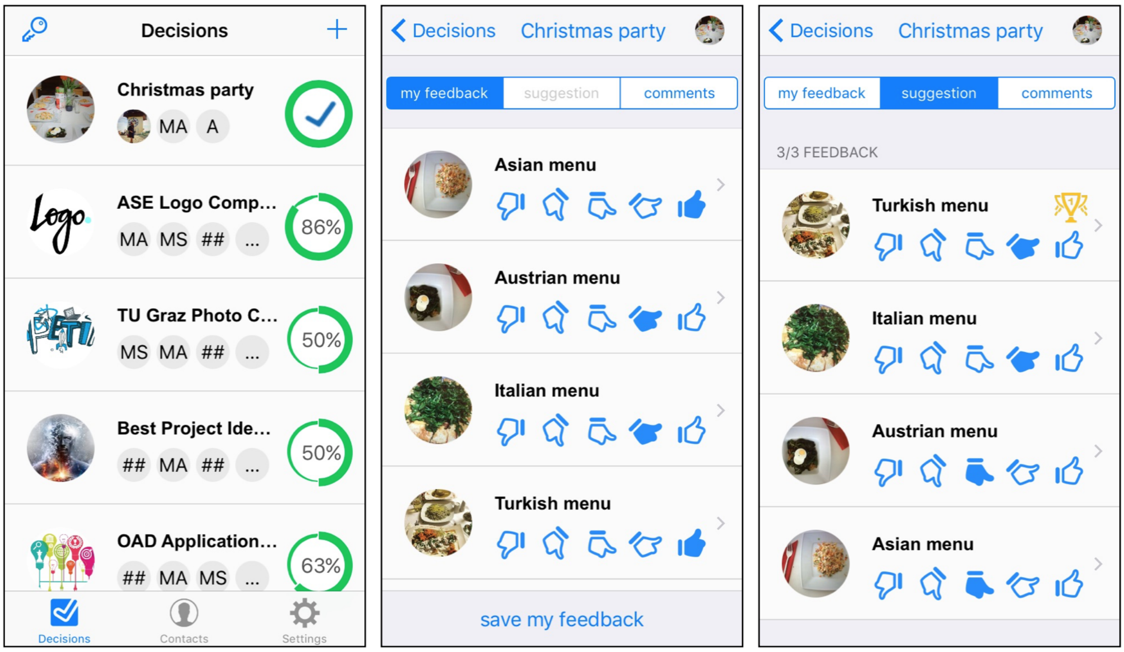
User	**Recipes**				
	$recipe_1$	$recipe_2$	$recipe_3$	$recipe_4$	$recipe_5$
$user_1$	4	4	4	2	5
$user_2$	4	4	5	4	5
$user_3$	5	2	5	4	3
$user_4$	4	5	3	3	4
Group (Least Misery strategy)	**4**	2	3	2	3

Table 7.7.: An example of using *Least Misery* strategy to aggregate individual user preferences into a group profile. $recipe_1$ is recommended to the group since its group rating value is highest.

Also, to support a group decision making process in a family, Elahi et al. (2014) proposed an innovative interactive environment for groups in planning their meals through a conversational process based on critiquing (Chen and Pu, 2012). The system consists of two components. The first one is a tagging and critiquing-based user interface. The second one is a utility function that takes into account the diet compliance and healthiness of the users. The utility for each meal is calculated on the basis of considering *meal time, user rating, diet plan*, and *health situation* of each group member. After that, the utility of each meal for the whole group is quantified by aggregating the individual utility scores of all group members. Based on the utility of each meal for the whole group, the system delivers a meal recommendation list for the group. Each group has a *group leader* (also called *the cook*), and *participants* who will attend the group meal. Sometimes, the cook must not select the recipe with the highest utility score. (S)He can accept or refuse recipe(s) for some reasons (e.g., the unavailability of ingredients or insufficient cooking-skills). The participants are allowed to criticize the meal, which was chosen by the cook. This critiquing process will be repeated until all members are satisfied.

Until now, to the best of our knowledge, there have been only a few research on food recommender systems for groups. In the mentioned study (Elahi et al., 2014), although proposing a new interactive mechanism for group in the food domain, it exposes many issues to be tackled in terms of group decision making, such as *bundle recommendation, fast consensus in a group, time of preference visibility, etc.* Figure 7.3 illustrates the user interfaces of the CHOICLA[¶] group decision support environment (Stettinger, 2014), which can be applied as a potential solution for supporting the group decision making process in the food domain. CHOICLA can support a group of friends to choose a menu for a Christmas party in an asynchronous fashion. That means all group members can join the decision making process without being on-line together at the same time. In this scenario, one member creates a decision (e.g., *Christmas party*) and enters some menus into the decision. Each menu is described by *name, photo, and description*. While joining in a decision, each group member can invite other members to participate in this decision. Invited members give their preferences by rating proposed menus (e.g., using thumbs up and thumbs down) and

[¶]The version of CHOICLA presented in this chapter was updated. Find the latest version of CHOICLA in www.choiclaweb.com



(a) (b) (c)

Figure 7.3.: Screenshots of the CHOICLA group decision support environment (iOS version). Figure 7.3(a) shows a list of different group decisions created by users. Users can rate alternatives by using the user interface shown in Figure 7.3(b). The suggestion for the whole group is shown in the *"Suggestion"* tab (Figure 7.3(c)). The alternative enclosed with the *medal icon* is the suggested alternative for the whole group. For instance, *"Turkish menu"* is chosen by CHOICLA to recommend to the whole group.

can discuss with each other using the *"comment"* functionality. Rating values from group members will be aggregated into group preferences using some group decision heuristics (e.g., *average, least misery, most pleasure, etc.*) (Masthoff, 2011) to propose a menu for the whole group. To avoid the anchoring effects (Felfernig, 2014), the group suggestion is solely shown to a group member after he/she saved the ratings for menus. Having said that, CHOICLA is the potential application for group decision processes in the food domain. However, the future version of CHOICLA should integrate a complete group decision process for the food domain, which takes into account additional information of all group members (such as *health situations, allergies, nutritional consumption, cooking skills, the availability of ingredients, etc.*), in order to recommend healthy food to the whole group.

7.5. Research challenges

Existing research on food recommender systems plays a crucial role in supporting users to choose a diet that suits interests and health conditions. These studies exploit information regarding user-profiles and recipes to generate food recommendations. It has been recognized that the recommendation quality is strongly influenced by the adequacy and accuracy of user information as well as nutritional information of food. However, recent studies have not provided detailed discussions on this issue. Besides, although some papers (e.g., (Ueta et al., 2011; Aberg, 2006)) propose food recommendations to tackle health problems, suggestions regarding changing eating behaviors, which are the premise to maintain a healthy lifestyle, are still missing. Explanations could help users more trust in recommendations and encourage them to follow good eating habits. However, the inclusion of explanations into food recommender systems has not received the interests of researchers. Besides, research on food recommender systems primarily focuses on single-user scenarios rather than on group scenarios. Until now, research on group recommender systems in the healthy food domain is quite limited. (Berkovsky and Freyne, 2010) is one of the studies which proposes some aggregation strategies to generate food recommendations for groups of users. However, there still exist some open issues which should be taken into account within the scope of future work, such as achieving fast consensus within the group or fostering fairness among group members. In this section, we will discuss the research challenges in food recommender systems and propose some potential solutions. A summary of open issues is presented in Table 7.8.

Research challenges	**Proposed solutions**
Collecting user information	Taking advantage of information about users' previous meals (Van Pinxteren et al., 2011).
	Implicitly collecting user information so that they don't have to invest too much time and effort (Freyne and Berkovsky, 2010).
Gathering nutritional information of recipe	The quantity of gathered recipes should be representative enough to vary the recommendations.
Recommendation algorithms	Improving the quality of recommendations by integrating constraints (e.g., health situations, nutrition needs, the availability of ingredients) into the recommendation process.
Explaining recommendations	Providing explanations which increase the trustworthiness of decision outcomes and persuade users to accept food recommendations (Elahi et al., 2014).
Changing eating behaviors	Employing health psychology theory in food recommender systems to encourage users to comply healthy eating behaviors (Snooks, 2009).
	Proposing potential dietary changes on the basis of exploiting the ideal nutrients from reliable resources (e.g., USDA, DACH).
Generating bundle recommendations	Expressing acceptable trade-offs among group members by employing negotiation and argumentation mechanisms (Felfernig et al., 2014b).
Achieving fast consensus in group decision making	Enriching user interfaces supporting basic negotiation mechanisms among group members (Nguyen and Ricci, 2017).

Table 7.8.: A summary of research challenges in food recommender systems and proposed solutions.

7.5.1. Challenges regarding User Information

- *The uncertainty of nutritional information from users*: In order to make recommendations, the system needs to collect *nutritional needs*, *ratings for food items/recipes* and *information of previous meals* from users (Mika, 2011). Most of the information is only provided through continuous interactions with users. However, in reality, recording nutritional intake from users can not avoid faults because users usually forget or give wrong information about the foods they have consumed (Mika, 2011). Although some systems were proposed to tackle with these problems, for instance, FOOD-LOG (Aizawa et al., 2010), they are not able to give the accurate information about the consumed meals, even though they can estimate the nutritional balance among different kinds of food in a meal.

- *Collecting user rating data:* Food recommender systems need information about users' preferences to recommend similar food items (Van Pinxteren et al., 2011; Mika, 2011). This information can be gathered by asking users to rate foods/recipes. However, it is not convenient if the system asks users to rate too many items. Hence, a challenge entailed is *"how to collect enough users' ratings while saving their effort?"* (Freyne and Berkovsky, 2010). Besides, similar to keeping reporting food consumption (as mentioned above), persuading users to keep rating dishes also becomes another challenge for food recommender systems (Mika, 2011).

7.5.2. Challenges regarding Recommendation Algorithms

As mentioned in (Mika, 2011), to calculate nutritional recommendations for users, any algorithm needs the following information:

- *User information* (e.g., likes, dislikes, food consumption, or nutritional needs): Similar to recommender systems in other domains, food recommender systems also face the cold-start problem when the system is used the first time (Mika, 2011). This problem can be surmounted by using information about users' previous meals to calculate similarity and then recommend new recipes to users (Van Pinxteren et al., 2011). However, this solution requires many user efforts and abates the desire for system usage.

- *Recipe databases*: Mika (2011) discussed two challenges that need to be solved:

  How many recipes the system should have? The number of gathered recipes should be large enough to accommodate the preferences of many users and vary the recommended recipes while still minimizing the time for making recommendations. This is a tricky problem when the system tries to balance between the variety of recommendations and system response time. Hoxmeier and Dicesare (2000) pointed out that long response times triggers user dissatisfaction which further decreases the continuous use of the system.

  How to gather accurate nutritional information of recipes? It is observed that with the same food item, if we use different ways to cook it, then we will get different nutritional values from it (Mika, 2011). Therefore, it is challenging to ensure that whether gathered nutritional tables for food items are accurate because when comparing different nutritional value table of foods, sometimes it returns varying values for the same food items (Mika, 2011). For instance, the nutritional value of *celery* in *'a salad recipe'* is different from the nutritional value of itself *'in a fried recipe'*, since cooking with high temperature make celery lose a significant amount of *essential oil*. It means that the amount of essential oil of celery in the '*fried recipe*' could be lower than in the '*salad recipe*'.

- *A set of constraints or rules*: Considering more constraints and rules in the recommendation process will improve the quality of recommendations (Mika, 2011). For instance, with a user who has heart disease, the system should recommend menus with less fat and salt. Moreover, it is indispensable to detect the conflicts among the constraints or rules which prevent the recommendation algorithms from finding a solution. However, with the extensive database (e.g., thousands of foods/recipes), checking constraints/rules in the database brings adverse effects for system performance (Mika, 2011). Besides, food recommender systems should take into account constraints concerning the availability of ingredients in households to help users save money and prevent food waste behavior. The challenge here is how to propose food which meets the health situations and nutritional needs of users, as well as taking advantage of the ingredients that are currently in their fridge. In this scenario, recommender systems seem to require many efforts from users because users have to report the consumption of all ingredients regularly, and this can prevent users from using the system permanently.

7.5.3. Challenges regarding Changing Eating Behavior of Users

Nowadays, many people are suffering from health problems because of inappropriate eating habits (Snooks, 2009). For instance, some people eat too much food compared to their physical activity level and gradually become obese. Whereas others (e.g., the elderly, the dieters) restrict extremely nutrition intake, and this leads to malnutrition. Therefore, one of the main functions of food recommender systems is to understand users' eating behaviors and to convince them to change their eating behaviors positively. However, this is a big challenge for food recommender systems since eating is a lifelong behavior that is influenced by many factors, especially psychological factors. Hence, food recommender systems should integrate *health psychology theory* in order to stimulate users to comply with healthy eating behaviors. The first approach can be used by applying a straightforward change at a specific time until the user behavior becomes habitual (Snooks, 2009). Another approach can be to compare to the ideal nutrient. Users can find the structure of an ideal diet according to their age and physical activity level from reliable resources (e.g., USDA, DACH) and then compare what food they ate to what is recommended (Snooks, 2009). The comparison approach was also proposed by Mankoff et al. (2002) to provide users potential dietary changes.

7.5.4. Challenges regarding Explanations

Explanations play a vital role in recommender systems since they increase the trust of users in decision outcomes (Tintarev and Masthoff, 2007). In the healthy food domain, explanations are even more essential since they not only increase the trust in recommendations but also stimulate users to consume healthy foods or to change their eating behaviors. For this purpose, it makes sense that explanations of food recommender systems clarify how a decision outcome is created (Elahi et al., 2014). Besides, a detailed description of food items (e.g., nutritional value table for a recipe) needs to be included in such a way that emphasizes the healthiness of specific food for users.

7.5.5. Challenges regarding Group Decision Making

As mentioned in previous sections of the chapter, recommending recipes/food items usually involves groups rather than individual users. However, there is a low amount of research on food recommender systems for groups. Therefore, it is still an open topic that needs to be analyzed in future research.

- *Bundle recommendations*: Group recommender systems usually attach the requirements/preferences of different users into group recommendation. This is the fundamental idea discussed in many related studies (O'Connor et al., 2001; Berkovsky and Freyne, 2010; Masthoff, 2011). In the food domain, the aggregation process raises more challenges when users want to get recommendations for a complete meal with the combination of many recipes/food or a food schedule for more than one day (e.g., foods for next week). This issue is known as *bundle recommendation* which is a new research branch of recommender systems. The idea here is to recommend a sequence of items instead of separated ones. In the healthy food domain, recommending a complete meal is even more complicated because the system has to consider not only the preferences of group members but also other aspects, such as *the variety of meals, weather and season* (Van Pinxteren et al., 2011), *the healthiness of recipes, health problems*, or *daily nutrition needs of group members*. On the other hand, the recommendation of bundles has to assure fairness among group members. This means negotiation and argumentation mechanisms have to be developed in order to support group members to express acceptable trade-offs (Felfernig et al., 2014b). For instance, in a meal plan for a week, the preferences of users who were discriminated in previous meals should have a higher emphasis on the upcoming meals.

- *Achieving fast consensus in groups*: In group recommender systems, although different aggregation approaches have been applied to generate group recommendations, such processes do not ensure that the recommended items reflect a high agreement level among group members (Castro et al., 2015). In this context, achieving consensus helps to bring individual preferences closer to each other before delivering group recommendations. However, further issues need to be considered in order to accelerate the achievement of consensus in groups. One of the promising solutions is to enrich user interfaces that support basic negotiation mechanisms among group members. User interfaces are designed such that all members can share their preferences within the group (Nguyen and Ricci, 2017). Knowing the preferences of each other helps the group to reach a consensus quickly. An example thereof is the following: user *A* prefers *"cheese"*, whereas user *B* is interested in *"beef"*. There is a probability of achieving a consensus between these two users is that user *A* would accept recipes with *beef* as long as they include *cheese*. How to represent the current decision situation is also considered as an issue of future work.

7.6. Conclusion

In this chapter, we provided an overview of recommender systems in the healthy food domain based on discussing four different types of food recommender systems. The first three types present some existing studies in the healthy food domain, which primarily focus on tailoring recommendations to individuals, by considering the preferences and/or nutritional needs of users. Meanwhile, recent studies presented in the fourth type target at consulting healthy food items in group scenarios. Popular recommendation approaches (e.g., collaborative filtering recommendation, content-based recommendation, constraint-based recommendation) are used in many food recommender systems. Besides, hybrid approaches are also employed to improve the recommender's performance. Although being considered in different contexts, in general, all food recommender systems play a vital role in providing food items that meet the preferences and adequate nutritional needs of users as well as convincing them to comply with positive eating behaviors. Some challenges regarding user information, recommendation algorithms, changing eating behaviors, and group decision making were discussed as open issues for further work.

Chapter 8

Investigating Serial Position Effects in Sequential Group Decision Making

This chapter is based on the results documented in (Tran et al., 2018b). All parts of this chapter in terms of literature research, user study, data analysis, and writing were done by the author of the thesis.

8.1. Abstract

Group decision making is performed in real life to select an optimal solution for the whole group. Decision making behavior of group members could be impacted by item domains and the chronological order in which decision tasks are presented to groups. In this chapter, we analyze situations where group members could apply different decision strategies depending on the chronological order of decision tasks. The analysis results confirm that item domains and the order of decision tasks have an impact on group decision strategies. This is especially the case where the preferences of a minority of group members are significantly different from other group members and when decision tasks related to high-involvement item domains are arranged before decision tasks in low-involvement item domains. Besides, we also figure out that group members invest different amounts of time in making a decision task depending on its position in a sequence of decision tasks.

8.2. Introduction

Group recommender systems can be regarded as tools that support group decision making processes. For instance, a critiquing-based recommender system (McCarthy et al., 2006) supports a group of friends to jointly plan a skiing destination for the Christmas vacation. A television program recommender system (Masthoff, 2004) allows a group of users to choose a sequence of television programs. An intelligent group recommender system so-called CHOICLA (Stettinger et al., 2015) provides an environment that enhances group coordination in completing joint decision tasks as well as improves the overall quality of group decision outcomes. Group recommendations are usually generated by aggregating the preferences of individual group members based on group aggregation strategies (Masthoff, 2011). The outcome of a strategy reflects the preferences of the whole group regarding a set of items.

The group decision making behavior of group members could be influenced by different factors, such as *decision making environments* (Chung and Adams, 1997), *decision tasks* (Reitz, 1977), *decision makers' characteristics* (Khasawneh and Abu-Shanab, 2013), and the *item type* (Felfernig et al., 2017). In the context of repeated group decision making, to some extent, it is still unclear whether there exists any influence of the order of decision tasks (in a sequence of decision tasks) on the chosen group aggregation strategy. In other words, it needs to be clarified whether there exist *serial position effects* (Felfernig et al., 2007a) that could unconsciously lead group members to different behaviors when making a sequence of group decisions. A realistic scenario could be defined as follows: A group of friends has to make different decisions which are arranged in a sequence of decisions corresponding to different domains. At the beginning, if the group is confronted with a decision on items with *low decision effort* (e.g., selecting a list of *songs* to be played in the fitness center in the next hours), the item that satisfies the preferences of the *majority* of group members could be chosen by the group, i.e., the preferences of a *minority* of group members could be ignored. However, if this decision is performed after a decision on items with *high related decision effort* (e.g., selecting an *apartment* to be shared for the next two years), then the decision making behavior of group members in the *song* domain could be unintentionally influenced by the behavior that the group already applied in the *shared-apartment* domain. This means the decision of which songs to choose could be based on the preferences of *all* individual group members (i.e., nobody is ignored) since the group most probably applied a similar behavior in the shared-apartment domain.

To the best of our knowledge, in-depth analyses of group decision making behavior depending on the order of decision tasks in a sequence of tasks do not exist. A related work presented in (Masthoff, 2004) analyzes the influence of viewing an item on giving the ratings for other items. However, this work solely focuses on items in a specific domain (e.g., television programs). An exploration of the group decision making behavior for a sequence of different domains has not been taken into account yet. In this chapter, we will go one step further by investigating serial position effects in a sequence of decision tasks with many different domains. We analyze *group aggregation strategies* (Masthoff, 2011) in order to figure out which strategy is applied by user study participants in which sequence of decision tasks. The awareness of this issue is the premise of *improving the prediction quality* of group recommender systems. Besides, we also examine the influence of the decision task order on the spending time of group decision making. This is achieved by measuring and analyzing time duration that participants need to solve decision tasks in a given sequence.

The remainder of the chapter is organized as follows. In *Section 8.3*, we briefly introduce the main idea of group aggregation strategies that are used to analyze the decision behavior of group members. In *Section 8.4*, we define hypotheses and present shortly main steps of our user study. The data analysis results as well as discussions regarding the proposed hypotheses are then presented in *Section 8.5*. In the last section, we conclude the chapter and discuss issues for future work.

8.3. Group Aggregation Strategies

Within the scope of our user study, we discover the decision making behavior of group members by analyzing various *group aggregation strategies* (Masthoff, 2011). There are two types of group aggregation strategies: *consensus-based aggregation* and *borderline aggregation* (Masthoff, 2004; Senot et al., 2017). These strategies are applied to merge the preferences of all individual group members into a group model that represents the inferred preferences of the whole group. In our research, we choose *Average (AVG)* and *Multiplicative (MUL)* as the representatives of the consensus-based strategies. Additionally, we selected *Least Misery (LMS), Most Pleasure (MPL), Minimal Group Distance (MGD),* and *Majority Voting (MAJ)* to be the representatives of the borderline aggregation strategies. A short description of applied aggregation strategies is presented as follows:

- *Average (AVG) recommends the item with the maximum average of individual ratings.*
- *Multiplicative (MUL)* recommends the item with the maximum product of individual ratings.

- *Least Misery (LMS)* recommends the item with the highest of all lowest individual ratings.
- *Most Pleasure (MPL)* recommends the item with the highest of all individual ratings.
- *Minimal Group Distance (MGD)* recommends the item which has the minimum distance to all individual ratings.
- *Majority (MAJ)* recommends the item with the highest number of all evaluations representing the majority of item-specific evaluations.

	holiday 1	*holiday 2*	*holiday 3*
user 1	3	1	5
user 2	3	4	2
user 3	3	4	2
user 4	3	4	2

Table 8.1.: A predefined group decision making scenario where group members explicitly evaluate different holiday destinations using a 5-star rating scale (1: the worst, 5: the best).

Strategies	*holiday 1*	*holiday 2*	*holiday 3*	**Recommendation**
AVG	3	**3.25**	2.75	*holiday 2*
LMS	**3**	1	2	*holiday 1*
MPL	3	4	**5**	*holiday 3*
MGD	3	**4**	2	*holiday 2*
MAJ	3	**4**	2	*holiday 2*
ENS	3	**4**	2	*holiday 2*
MUL	**81**	64	30	*holiday 1*

Table 8.2.: On the basis of the group members' evaluations for the holiday destinations (see *Table* 8.1), group aggregation strategies are used to recommend a corresponding holiday to the group.

Besides, we also use the *Ensemble voting (ENS)* strategy to determine the majority of the results of all individual voting strategies. *MUL* is not taken into account by the *ENS* strategy since its output lies within a different value range (i.e., not from 1 to 5). A more detailed discussion of the computation functions of the aforementioned aggregation strategies is given in (Felfernig et al., 2018a). An example of the application of these strategies is shown in Table 8.2.

8.4. Hypotheses and User Study

8.4.1. Hypotheses

The main goal of our study is to answer the following research question: *'Do serial position effects occur in the context of group decisions which are performed in different domain sequences?'*. In this context, we tried to test the following two hypotheses (H_1 and H_2):

- H_1: *"User study participants are assumed to apply different group aggregation strategies for the same decision task depending on its position in the given sequence of decision tasks"*.

  The motivation of H_1 is to figure out the tendency to reuse previously applied strategies when making sequential group decision tasks.

- H_2: *"In the context of repeated group decision making in a sequence of different decision tasks, user study participants are assumed to invest different amounts of time for the same task depending on its position in a sequence of decision tasks".*

  The motivation of H_2 is to prove the fact that different strategies with regard to constructing sequences of decision tasks require different time efforts. Besides, this hypothesis helps to better compare the difference between decision task sequences concerning the total time needed to complete all the tasks in a given sequence.

8.4.2. User study

Our user study was designed and performed in the following three steps: (1) Define evaluation settings, (2) Construct decision tasks from evaluation settings, and (3) Build sequences of decision tasks with regard to different item domains and deliver to user study participants.

Step 1 - Define evaluation settings

We assumed a situation in which four imaginary group members already rated three items using a *5-star* rating scale. The ratings of the four group members about an item were represented as a setting with four ratings. In our user study, we chose the following *five* settings ($S_1 - S_5$) to describe different situations of group members' preferences (see Table 8.3):

- S_1 - *Average support - AVS (3, 3, 3, 3)* represents a situation where each group member provides an average rating for the item and the ratings of group members are the same.

- S_2 - *Disagreement - DIS (1, 2, 3, 4)* describes a situation where group members do not show a clear opinion about the item and their item ratings range from negative to positive.

- S_3 - *Majority positive - MJP (1, 4, 4, 4)* represents a situation where a majority of group members like the item; only a minority of group members do not like the item.

- S_4 - *Majority negative - MJN (5, 2, 2, 2)* represents a situation where a majority of group members do not like the item; only a minority of group members like the item.

- S_5 - *Polarization - POL (4, 4, 1, 1)* describes the situation where there exist two different opinion flows on the item (one-half of the group members supports the item, another half does not support the item).

Setting	Setting name	*1st* user	*2nd* user	*3rd* user	*4th* user
S_1	AVS - Average support	3	3	3	3
S_2	DIS - Disagreement	1	2	3	4
S_3	MJP - Majority positive	1	4	4	4
S_4	MJN - Majority negative	5	2	2	2
S_5	POL - Polarization	4	4	1	1

Table 8.3.: Settings of user preferences (evaluations) used in the user study (S_1 - S_5). Preferences are expressed in terms of ratings on a 5-star rating scale (1: the worst, 5: the best)

Step 2 - Construct decision tasks from evaluation settings

We defined *ten different decision tasks* (*Task 1, Task 2, ..., Task 10*) in which each task was tailored by combining *three out of the five* mentioned settings ($\binom{5}{3} = 10$) (see Table 8.4). An example decision task is shown in Table 8.1. This task was constructed by three evaluation settings *(3,3,3,3), (1,4,4,4),* and *(5,2,2,2)* corresponding to *holiday 1, holiday 2,* and *holiday 3* respectively. Within each task, settings were shown to user study participants in a randomized fashion. Each participant was asked to select an item from the set of three items for which the group ratings were provided.

Task	1^{st} setting	2^{nd} setting	3^{rd} setting
1	S_1	S_2	S_3
2	S_1	S_2	S_4
3	S_1	S_2	S_5
4	S_1	S_3	S_4
5	S_1	S_3	S_5
6	S_1	S_4	S_5
7	S_2	S_3	S_4
8	S_2	S_3	S_5
9	S_2	S_4	S_5
10	S_3	S_4	S_5

Table 8.4.: Tasks used in the user study. The settings were taken from Table 8.3. $M = \binom{5}{3} = 10$ tasks represent all possible combinations of three out of five settings.

Step 3 - Build sequences of decision tasks with regard to different item domains and deliver to user study participants

Each sequence consisted of *four decision tasks* related to *four item domains*. Different participants received different sequences. Within sequences, decision tasks corresponding to different item domains were shown to participants in random orders. We chose four item domains: *very-low-involvement, low-involvement, high-involvement,* and *very-high-involvement*. A *(very)-low-involvement* item domain includes items with *(very)-low decision making effort* in terms of price, risk factor, and decision making effort. In contrary, a *(very)-high-involvement* item domain includes items with *(very)-high decision making effort*. As a very-low-involvement item domain, we chose the *music genre* domain where a collection of songs from the chosen music genre will be played in a fitness center for the next two hours. As a low-involvement item domain, we chose the *restaurant* domain where a group of users has to decide on a restaurant for the upcoming dinner. As a high-involvement item domain, we chose the *holiday* domain where a group of friends has to select a destination for the next summer vacation. As a very-high-involvement item domain, we chose the *shared-apartment* domain where a group of students has to decide on an apartment to be shared in the next couple of years.

We conducted our study with students from three Austrian universities.* In total, there are $N = 305$ participants (*males*: 193, *females*: 112) who had to individually select items in the mentioned domains. We want to emphasize that user study participants were not group members involved in decision tasks pre-defined in *Step 2*. They played the role of consultants who analyzed a given decision task and selected an item that was assumed to be optimal for the group. We used the resulting dataset (i.e., recommendations of group

*Graz University of Technology (www.tugraz.at), Karl-Franzens University of Graz (www.uni-graz.at), and University of Klagenfurt (www.aau.at) decisions given by participants) to evaluate the *prediction quality* (i.e., *precision*) of different group aggregation strategies. The precision of an aggregation strategy was measured in terms of *the ratio between the number of correctly predicted group decisions and the overall number of predictions*.

8.5. Data Analysis Results and Discussions

In our user study, the four chosen domains and related decision tasks were arranged into different sequences with different orders and categorized into two types. HIGH → LOW: (very)-high-involvement item domains were presented to participants before (very)-low-involvement item domains. LOW → HIGH: (very)-low-involvement item domains were shown to participants before (very)-high-involvement item domains. Possible sequences of each type are shown in Table 8.5.

Sequence type	Possible domain sequences
HIGH → LOW	H-A-M-R, H-A-R-M, A-H-M-R, A-H-R-M
	H-M-A-R, H-R-A-M, A-R-H-M, A-M-H-R
LOW → HIGH	M-R-H-A, M-R-A-H, R-M-H-A, R-M-A-H
	M-H-R-A, M-A-R-H, R-H-M-A, R-A-M-H

Table 8.5.: Possible sequences of four chosen domains. For instance, H-A-M-R denotes a sequence with the following order: *1st*: H (holiday), *2nd*: A (shared-apartment), *3rd*: M (music genre), and *4th*: R (restaurant).

Hypothesis H_1 - *"User study participants are assumed to apply different group aggregation strategies for the same decision task depending on its position in the given sequence of decision tasks."*

In order to test this hypothesis, first we collected all possible sequences which belong to a sequence type (i.e., HIGH → LOW or LOW → HIGH). After that, within each item domain, recommendations of participants related to decision tasks that have been assigned to the domain were collected. The recommendation of a participant for a decision task is an item chosen from three items in the given decision task. This item was considered by the participant as the best solution for the group. A dataset of participants' recommendations in ten different tasks was analyzed for the purpose of investigating tasks where serial position effects really exist. In each task, the precision of aggregation strategies was calculated. The data analysis results show that serial position effects occurred in *Task 4* which were constructed from three evaluation settings (*(3,3,3,3), (1,4,4,4),* and *(5,2,2,2)*). Task 4 represents a scenario in which, *for most items, the preferences of one group member are different from the preferences of other group members*. We recognize that there exists a significant change in the group decision making behavior of user study participants on this task. Serial position effects observed on this task are the following:

HIGH→LOW sequences: In these sequences, the participants invested more time for making decision on *Task 4* when it was delivered to the participants in *high-involvement item domains* (see Table 8.6). In addition, the participants applied *MUL* and *LMS* strategies to generate group recommendations (i.e., *MUL* and *LMS* strategies achieved the highest precision both in the *holiday* and *shared-apartment* domains) (see the upper part of Table 8.7). The precisions of the two strategies are the same since both of them recommend the same item to groups. Such group decision making behaviors could be explained as follows: *MUL* is recognized to perform most effectively in the context of group decision making on a sequence of items (Masthoff, 2011), and this could also be the case for a sequence of domains. Besides, *LMS* makes sense in the context of high-involvement item domains because it considers the preferences of the least satisfied group member (i.e., the group member who gave the lowest rating for the item) and this helps to minimize misery within the group. In addition, spending more time on decision tasks related to high-involvement item domains could make participants stick with these decisions. Therefore, they tended to use the same strategies for the follow-up decision tasks in low-involvement item domains. The results in the upper part of Table 8.7 show that participants used the same strategies (i.e., *LMS* and *MUL*) to make group decisions for the *music genre* and *restaurant* domains. In other words, in situations where low-involvement items follow high-involvement items, there is a tendency to *keep the decision strategy* used for high-involvement items. Moreover, we also figure out that reusing decision strategies applied in high-involvement item domains for the follow-up decision tasks in low-involvement item domains could help to improve the prediction quality of group recommender systems (e.g., the precision of *LMS* and *MUL* strategies significantly improved in the *restaurant* domain - see Table 8.7).

Domain	**Average time consumption** (seconds)
holiday	106.1
shared-apartment	65.9
music genre	41.1
restaurant	48.9

Table 8.6.: The distribution of the average time consumption of the participants for *Task 4* in different item domains.

LOW→HIGH sequences: In these sequences, the group decision making behavior of participants in high-involvement item domains is not influenced by the behaviors performed by the participants in low-involvement item domains. In other words, high-involvement items seem to *trigger a switch* in the decision strategy. This fact is confirmed by the data analysis results depicted in the lower part of Table 8.7. Participants applied *LMS* and *MUL* strategies for items in low-involvement domains, whereas *AVG*, *MGD*, *MAJ*, and *ENS* strategies were used for in high-involvement item domains. Moreover, the prediction quality of *AVG, MGD, MAJ*, and *ENS* in the *holiday* and *shared-apartment* domains is improved significantly whenever group decisions in these domains are performed after group decisions in (very)-low-involvement item domains have been made.

To conclude, the hypothesis H_1 can be partly confirmed for HIGH→LOW sequences which consist of tasks in which the preferences of a minority of group members for most items significantly differ from the preferences of other group members.

Hypothesis H_2 - *"In the context of repeated group decision making in a sequence of different decision tasks, user study participants are assumed to invest different amounts of time for the same task depending on its position in a sequence of decision tasks."*

To examine the hypothesis H_2, we measured the time that participants invested in the decision tasks in the four chosen domains. For each domain, time durations of all tasks for HIGH→LOW and LOW→HIGH sequences are collected into different sets. Time durations in these sets are normalized using Formula 8.1. After that, we used the *Independent t-test* (*significant level* $\alpha = 0.05$) to determine whether there exists a statistically significant difference between the population means from two different sets of time duration.

$$norm\text{-}duration = \frac{duration - duration_{min}}{duration_{max} - duration_{min}} \tag{8.1}$$

The t-test analysis results on different sets of time duration obviously show that: Compared to LOW→HIGH sequences, in HIGH→LOW sequences, decision tasks related to *high-involvement item domains* (i.e., *holiday* and *share-apartment*) take longer ($P_{holiday}$ *one tail* = $0.00624 < \alpha$ and $P_{shared-apartment}$ *one tail* =

Domain	AVG	LMS	MPL	MGD	MAJ	ENS	MUL
			HIGH $\rightarrow$ LOW				
holiday	25%	**75%**	0%	25%	25%	25%	**75%**
shared-apartment	**50%**	**50%**	0%	**50%**	**50%**	**50%**	**50%**
music genre	25%	**75%**	0%	25%	25%	25%	**75%**
restaurant	12.5%	**87.5%**	0%	12.5%	12.5%	12.5%	**87.5%**
			LOW $\rightarrow$ HIGH				
music genre	37.5%	**62.5%**	0%	37.5%	37.5%	37.5%	**62.5%**
restaurant	31.25%	**62.5%**	6.25%	31.25%	31.25%	31.25%	**62.5%**
holiday	**62.5%**	37.5%	0%	**62.5%**	**62.5%**	**62.5%**	37.5%
shared-apartment	**50%**	25%	25%	**50%**	**50%**	**50%**	25%

Table 8.7.: An analysis of the precision (i.e., the prediction quality) of group aggregation strategies in two types of domain sequences of *Task 4*. *Task 4* represents a scenario in which, for most items, the preferences of a minority of group members differ from those of other group members (see Table 8.1).

$0.033 < \alpha$). The opposite tendency is confirmed in decision tasks related to *low-involvement item domains*. That means, compared to the HIGH$\rightarrow$LOW sequences, in LOW$\rightarrow$HIGH sequences, participants tend to spend more time for decision tasks in *low-involvement item domains* (P_{song} *one tail* $= 3.75E-08 < \alpha$ and $P_{restaurant}$ *one tail* $= 4.6E-05 < \alpha$).

To conclude, in the context of repeated group decision making, the hypothesis H_2 is supported. In other words, with the same decision task, the duration for making a decision could differ depending on which position it appears in a sequence of different item domains. Besides, the confirmation of the hypothesis H_2 helps to figure out that participants tend to invest more time for the tasks arranged in the first positions of the sequence of decision tasks and less time for the ones which appear at the end of the sequence.

8.6. Conclusions and Future Work

In this chapter, we focused on analyzing the existence of serial position effects in sequential group decision making. The results of our study confirm that the decision making behavior of group members changes significantly when high-involvement item domains are shown before low-involvement item domains. This fact is also confirmed whenever group members are confronted with a decision task where the preferences of a group member concerning most items significantly differ from the preferences of other group members. In such decision tasks, group members tend to reuse the strategies already applied to high-involvement item domains for the follow-up decisions in low-involvement item domains. The reusing tendency helps to improve the prediction quality of group recommender systems. Additionally, we figured out that there exist serial position effects on the group decision making duration of group members. In repeated group decision making scenarios, group members invest different amounts of time for the same decision task depending on its position in a given sequence of decision tasks.

Within the scope of our future work, we will focus on the analysis of group decision strategies by comparing their item domain-specific sensitivity. For example, we will analyze the impact of integrating different aspects of *risk-awareness* into the design of group aggregation strategies. This helps to figure out which group aggregation strategy is optimal in terms of minimizing risk or misery within group members. Besides, we will repeat this user study by creating real groups (instead of artificial groups) and investigate the influence of different aspects that could occur in the group decision making process (e.g., *age, gender, cultural background, social influence, dominant players, etc.*).

Chapter 9

Conclusions & Future Work

In daily life, groups of users are usually engaged in numerous decisions, such as deciding on a destination to visit or choosing a house to buy (Garcia et al., 2009). In such scenarios, group recommender systems are beneficial to support group decision making processes. The thesis proposes different decision support techniques that help to increase the efficiency of group decision making processes and to improve the quality of decision outcomes. In this chapter, we summarize the thesis contributions in the form of corresponding answers to research questions and then discuss the limitations of our approaches as well as some open issues for future work.

9.1. Conclusions

This section summarizes the answers to the research questions defined in Section 1.2.

Research Question Q1.1:

How to support configuration for groups?

To support the configuration process for groups of users, we proposed a new configuration approach denoted as *group-based configuration* which allows a group of users to jointly configure complex products/services. In *Section 2.3*, we introduced definitions of a group configuration task and a corresponding solution. These definitions are based on a *Constraint Satisfaction Problem* (CSP) which is often used to define single-user configuration tasks. Differing from other types of group decision tasks, the major characteristics of group-based configuration are defined in terms of a knowledge base, i.e., the alternatives are not pre-specified. Our approach provides a new mechanism to configuration and diagnosis, and the configuration task that needs to be represented in a corresponding user interface.

Research Question Q1.2:

How to resolve inconsistencies in group-based configuration scenarios?

In group-based configuration scenarios, there could exist *inconsistencies* which cause the *"no solution can be found"* dilemma. The inconsistencies are usually triggered between two group members' preferences or between group members' preferences and the knowledge base. In situations where inconsistencies exist, a minimal set of constraints (also called a *diagnosis*) that causes conflicts has to be manually adapted/deleted by group members. This way, we could always find at least one solution for the group. In *Section 2.4*, we proposed a solution to deal with inconsistent preferences of group members based on the concepts of *model-based diagnosis* (Reiter, 1987). This solution showed how different types of *preference aggregation heuristics* (Masthoff, 2011; Felfernig et al., 2018a) can be integrated into the diagnosis process.

Research Question Q2:

How to better detect hidden profiles of group members in group-based configuration?

In group-based configuration, knowledge gaps of some group members result in a low quality of the preference acquisition process. Therefore, in this context, advice from domain experts can be beneficial for group members to evaluate products and services more precisely. To answer this research question, we proposed a new approach based on the concept of *liquid democracy* (Blum and Zuber, 2016; Boldi et al., 2015). This approach allows group members to either evaluate the items or delegate their rating power to domain experts (see *Section 3.4*). Besides, regarding the application of liquid democracy, in *Section 3.5* we presented a novel approach based on *Multi-attribute Utility Theory (MAUT)-based evaluation* (Dyer, 2005) to calculate the utility of configurable items. The *MAUT-based evaluation* method assigns a higher importance to the domain experts and also takes into account their expertise levels when calculating the utility of items.

Research Question Q3.1:

How to explain recommendations for groups?

Sending group recommendations to users in the form of *"black boxes"* prevents them from understanding the underlying mechanism of the recommendation process. As a result, it can lead to users' skepticism (Bilgic and Mooney, 2005). In this context, explanations for recommended items are very helpful for users to give an insight into the group recommendation process, to make better decisions, and to increase their acceptance with regard to recommended items (Tintarev and Masthoff, 2007; Felfernig et al., 2018d). In this thesis, we discussed different approaches to explain group recommendations in the context of basic recommendation paradigms such as *collaborative filtering* (*Section 4.4*), *content-based filtering* (*Section 4.5*), *constraint-based* (*Section 4.6*), and *critiquing-based recommendation* (*Section 4.7*). Especially, these approaches also take into account specific aspects of group recommendation scenarios, such as *fairness* and *consensus* among group members. Moreover, for each recommendation paradigm, we also proposed some *verbal explanations* and corresponding *visualizations* to give a more in-depth comprehesion of how explanations can be determined.

Research Question Q3.2:

How can explanations help to increase the fairness and consensus perception of users with regard to group recommendations?

Group recommender systems should consider social aspects among group members to maximize the satisfaction of users with recommended items. Some examples of the social aspects could be *"fairness"* which takes into account as far as possible the preferences of group members and *"consensus"* which convinces group members to agree on a decision (Felfernig et al., 2018d). For this research question, we proposed different types of explanations which aim to increase the fairness and consensus perception of users concerning group recommendations (see *Section 5.4.2*). The proposed explanations are denoted as *social choice-based explanations* describing the underlying mechanisms of preference aggregation strategies and taking into account the satisfaction of group members in previous or future decisions. We conducted a user study to evaluate the explanations according to three dimensions: *the fairness perception, the consensus* *perception,* and *the satisfaction* of group members with regard to recommended items (see *Section 5.5.2*). The experimental results show that the explanations describing aggregation strategies taking into account the preferences of *all* or the *majority* of group members performed the best in terms of the three mentioned dimensions (see *Section 5.6.1*). Besides, we detected *positive correlations* between the fairness/consensus perception and the satisfaction of users concerning group recommendations. Indeed, the results summarized in *Section 5.6.2* confirm that higher levels of perceived fairness/consensus of the explanations correlate with higher levels of satisfaction of group members with regard to group recommendations. On the other hand, in the context of repeated decisions, the integration of group members' satisfaction in previous decisions into the explanations can help to significantly increase the fairness perception of users. This tendency was manifested in explanations which describe a group recommendation strategy taking into account the preferences of a *subset* of group members (see *Section 5.6.3*).

Research Question Q4:

How to counteract decision manipulation in group recommender systems?

In group recommender systems, a decision manipulation indicates an attack where a group member tries to adapt the rating of items to push his/her favorite options (McCarthy and Anagnost, 1998; Jameson, 2004). Consequently, decision manipulation can trigger *"insincere"* user preferences and therefore can lead to low-quality decision outcomes. To counteract the negative influence of decision manipulation in group recommender systems, we proposed a UI-driven solution which makes group members' rating adaptations transparent. This solution was proposed based on the *Hawthorne Effect* (Sedgwick and Greenwood, 2015) indicating that *"users tend to avoid decision manipulation if they know their rating adaptations are seen by others"*. In *Section 6.4.2*, we proposed different UIs representing the rating adaptations of group members at different transparency levels. The proposed UIs were categorized into two groups. *Group 1* consists of UIs showing the information of who has adapted the rating of items, whereas *Group 2* includes UIs without showing this information. The empirical results presented in *Section 6.6.2* confirm the Hawthorne Effect in the context of decision manipulation. Additionally, the results also point out that the UI presenting the rating adaptations of group members at the highest transparency level most effectively counteract decision manipulation. Besides, the ability of UIs to counteract decision manipulation turns out different according to the information represented in the UIs. Indeed, the empirical results showed that UIs showing *group members* who have adapted the ratings of items better helped to discourage users from decision manipulation compared to those excluded this information. Furthermore, the information regarding *item ratings* and *recommended items* also effectively helped to prevent users from decision manipulation.

Research Question Q5.1:

How to generate food recommendations to groups of users?

The increase of available clinical data, which represents patients' health states, has risen the needs of using recommender systems to provide helpful information to users (Wiesner and Pfeifer, 2014). For instance, a patient who has diabetes might need recommendations concerning health-care services that show which medical remedies or food items that he/she should take. In this context, *health recommender systems* (Wiesner and Pfeifer, 2014; Schäfer et al., 2017) have emerged as useful tools to help users better understand their health situations and encourage them to follow healthier routines. In this thesis, we chose *food recommender systems* as an example of health recommender systems and discussed some approaches of how to generate recommendations in this domain (Freyne and Berkovsky, 2010; Ueta et al., 2011; El-Dosuky et al., 2012; Kuo et al., 2012; Elahi et al., 2014; Elsweiler et al., 2015). In *Section 7.4*, we presented different approaches to generating food recommendations for single users by taking into account various criteria, such as the *preferences, nutritional needs, health problems,* and *eating behaviors of users*. Especially, in *Subsection 7.4.4*, we presented some recommendation techniques for groups based on the *aggregation strategies* of individual group members' models (i.e., *aggregated models* and *aggregated prediction*).

Research Question Q5.2:

Which open issues in the healthy food domain should be taken into account within the scope of future work?

Recommender systems in the healthy food domain are an on-going field of research and there still exist many aspects that should be taken into account. In *Section 7.5*, we discussed some research challenges in food recommender systems with regard to *collecting user information, recommendation algorithms, changing users' eating behavior, and explanations for recommended food items*. Besides, we presented open issues in food recommender systems for groups (such as *bundle recommendations, fast consensus in a group,* or *time of preference visibility*) and proposed some potential solutions to resolve them.

Research Question Q6:

How do serial position effects influence the decision making behavior of group members in the context of sequential group decision making?

Serial position effects are usually referred to as decision biases triggered when items are represented in the form of a list (Mandl et al., 2011). In this thesis, we investigated another type of serial position effects occurring when a group of users makes different decision tasks continuously. In particular, we examined whether group members apply different *preference aggregation strategies* (Masthoff, 2011; Felfernig et al., 2018a) to the same decision task depending on its position in a given sequence of decision tasks. The empirical results in *Section 8.5* show that the decision making behaviors of users significantly change when decision tasks in high-involvement item domains are shown before those in low-involvement item domains. In particular, the decision making behavior of group members for decision tasks in low-involvement item domains tend to be influenced by the behaviors that have been applied to decision tasks in high-involvement item domains. We confirmed this effect in decision tasks where the preferences of a group member for the majority of items are significantly different from others'. In such decision tasks, users tend to re-use the strategies which were previously applied to decisions in high-involvement item domains for the follow-up decisions in the low-involvement item domains. On the other hand, in repeated group decisions, the time duration to make a decision task differs depending on its position in a given sequence of decision tasks.

Limitations

One of the limitations of the thesis lies in *the group decision scenarios*. Our conducted user studies were subject to analyzing group decision scenarios in *small-size* groups (e.g., from *three* to *five* group members). Additionally, we primarily focused on *homogeneous* groups where group members have similar backgrounds and know each other (e.g., a group of friends/colleagues/family members). However, big-size and heterogeneous groups (i.e., group members with different backgrounds and having no relationships with each other) have not been discussed. Moreover, we have not adequately considered the impacts of *group dynamics* on group decision making. Group dynamics refer to a system of behaviors and psychological processes occurring within group settings (Forsyth, 2006). Existing studies in social sciences have shown that group decision making processes are not always rational. Decision outcomes cannot always be generated by solely considering the preferences of individual group members. Other than that, features such as *group size, group members' characteristics, and group cohesiveness* can influence the quality of decision outcomes. As a result, group dynamics should be further analyzed to better support the group decision making process and boost the quality of group decisions (Felfernig et al., 2018c).

9.2. Future Work

Based on the limitations mentioned in *Section 9.1*, we propose open issues that need to be taken into account within the scope of our future work.

9.2.1. The Influence of Group Dynamics on Group Decision Making

Group size

Group size directly influences the outcomes of group decision making processes (Hackman and Vidmar, 1970; Desanctis and Gallupe, 1987; Chung and Adams, 1997). In small groups, users have adequate opportunities to express their opinions and seek clarifications on unclear points. Hence, it is easier for them to reach an agreement on a solution. In contrast, in larger groups, the probability of reaching a consensus is quite limited (Thomas and Fink, 1963; Shaw, 1976). In big groups, the discussions among group members are essential to make the right decisions. These discussions help users to integrate a wide range of knowledge, skills, and expertise from different areas. For instance, when developing a new software application, a decision involving a large group of stakeholders should be made. The decision takes into account the knowledge/skills of individuals coming from different areas, such as software developers, UI designers, project managers, customers, marketing representatives, financial employees, and customers. However, when group size increases beyond a certain point, the quality of the decision made by the group might decrease accordingly (Thomas and Fink, 1963; Shaw, 1976). This means the larger the group, the lower the cohesiveness, and the larger the potential conflicts. As a result, the group decision making process could last longer. For future work, we will further analyze the *social aspects* within groups with different sizes. For instance, we will try to investigate the *fairness perception, consensus perception*, *satisfaction* of users with regard to recommended items in different group sizes. We would argue that in small groups, users might achieve higher scores of the mentioned dimensions compared to in big groups. Besides, we will investigate at which group size the quality of group decisions reaches its maximum in terms of the mentioned dimensions.

Group cohesiveness

Group cohesiveness describes a social relation where group members interact with each other and generate the forces that push group members closer to each other (Piper et al., 1983; Beal et al., 2003). In group decision making, the cohesiveness of group members can influence how the group decision is made. For future work, we will investigate the decision making behavior of group members by analyzing the social relations among group members. These relations will be analyzed in different cohesiveness degrees ranging from *high cohesiveness* (e.g., groups of family members or long-time friends) to *low cohesiveness* (e.g., groups of unacquainted individuals). Particularly, we will try to examine the following assumptions (Thompson et al., 1998):

1. Cohesive groups are assumed to be subject to supportive communications in which group members are more comfortable to express their thoughts and their feelings compared to non-cohesive groups.

2. Users in cohesive groups are assumed to be more friendly and more cooperative with each other compared to users in non-cohesive groups.

3. Users in cohesive groups are assumed to show higher levels of social awareness (e.g., perceived fairness and perceived consensus) and the satisfaction with recommended items compared to users in non-cohesive groups.

Age difference

In group decision making, group members with different ages do not have the same behavior to each other when making a decision. Indeed, Sanz de Acedo Baquedano et al. (2007) surveyed 589 participants who are from 18 to 80 years old to explore the impact of *age diversity* on the decision making process. The empirical results show that youths tend to face pressure from environmental emotions and social aspects in their decisions compared to adults and retired persons. This is due to the lack of knowledge and experience in certain decision areas of the youths. Whereas adults and retired persons consider these factors more closely and evaluate the quality of their decisions after considering the appropriate strategies (Sanz de Acedo Baquedano et al., 2007). In another research, Thornton et al. (1997) stated that *the more significant the differences among individual users in a group, the higher the required compromise*. These findings lead to the following hypotheses that need to be examined in the future (Thornton et al., 1997):

1. The older the group members, the higher the quality of group decisions in terms of increasing social awareness within the group (e.g., the fairness or consensus perception of group members concerning group recommendations).

2. The age difference of group members could trigger more discussions within the group to reach a consensus.

3. The age difference of group members could trigger a longer group decision making process.

Gender difference

One of the common questions that could be raised in the context of group decision making is: *"Does the gender difference of group members have an impact on group decision outcomes?"*. Hannagan and Larimer (2010) confirmed that women and men use different strategies in the group decision making process. Indeed, in political areas, male and female legislators show different behaviors in debating and negotiating (Kathlene, 1994). Females tend to follow a more democratic style of leadership, whereas males are more likely to use an autocratic leadership style (Eagly and Johnson, 1990). In some research related to decision making, females are recognized more concerned with uncertainty, easily dominated by emotional factors, and impacted by the environment (Sanz de Acedo Baquedano et al., 2007; Khasawneh and Abu-Shanab, 2013). Besides, females are likely to consume more time in the decision making process and look for more details and information (Khasawneh and Abu-Shanab, 2013). In contrast, males focus on information analysis to carry out the decision. They tend to be more objective, realistic, assertive, and dominant (Sanz de Acedo Baquedano et al., 2007). Besides, males are more competitive and usually use the *winner-and-loser* approach during decision making processes, whereas females tend to use collaborative and cooperative strategies (Hannagan and Larimer, 2010).

Based on those mentioned features, the gender difference of group members could have a significant influence on the outcome of group decisions (Zaidi et al., 2010). Khasawneh and Abu-Shanab (2013) surveyed students at Yarmouk University (Jordan) to measure the impact of group members' gender on group decision making performance. Empirical results show that *female-only groups* have a better performance than *male-only groups* in terms of good ideas exchanged among group members. In the fund management area, the inclusion of males in a group can increase the probability of choosing a higher risk investment and decrease the probability of choosing a larger loss of investment (Borgan et al., 2012).

Within the scope of our future work, we further analyze the impact of the gender difference of group members on the social aspects discussed in the thesis. For instance, we will investigate whether gender diversity does affect fairness and consensus perceptions concerning recommended items. To examine this assumption, we will conduct a user study with different types of groups, such as *all-male* groups, *all-female* groups, and *mixed* groups with both males and females.

Cultural difference

In the era of globalization, it is vital to understand the influences of national culture on *multi-national* projects as well as the performance of decision making. In multi-national companies, there is a high probability of decision making processes that could be performed by groups of users who come from different cultures or religions. "*Users in a multi-national group can bring diverse backgrounds, experiences, and world perspectives that constitute a fertile knowledge for resolving complex problems*" (Rodriguez and Brodbeck, 2008). However, the cultural difference could result in a low level of cohesion among group members. Therefore, this could trigger a low decision making performance in terms of process speed, low decision quality, and higher levels of group conflict (Rodriguez and Brodbeck, 2008). Müller et al. (2009) conducted a user study to explore cultural differences in *decision making styles* in project groups formed by users who come from different nationalities (in this paper, German and Swedish). The results show that the decision making process of Swedish groups seems to be slow, more transparent, and less formal. Meanwhile, German groups are faster in decision making and assign more specific responsibilities to group members. Besides, German groups tend to be more dominated by the decision authority of an expert in the domain.

For future work, we will extend our user studies with culturally-heterogeneous and diverse groups to investigate the influence of cultural differences on group decision outcomes. For instance, we will survey groups where users have a different cultural or religious backgrounds. Thereafter, *social aspects* within the groups (e.g., *fairness* and *consensus*) will be analyzed by considering the cultural differences of group members.

9.2.2. Explanations and Visualizations for Supporting Consensus

In group recommender systems, before proposing a solution to the group, a consensus making process should be carried out to make individual group members' preferences closer to each other (Chiclana et al., 2014). The consensus making process can help to increase the satisfaction of group members with regard to group recommendations. However, this process is time-consuming since it is usually repeated until all conflicts among group members are resolved and the whole group agrees on a solution.

Therefore, within the scope of future work, we will propose some explanations and visualizations to accelerate and facilitate the consensus making process. These techniques could help users efficiently detect the conflicts amongst group members' preferences and recommend some solutions to solve them (Alonso et al., 2010). For instance, to show the *consensus state* of the group, some example explanations could be: *"At the moment, the consensus level of the group is still deficient since the preferences of users A, B, and C quite differ from each other"* or *"the consensus making process stops now since the current consensus level of the group has reached the pre-defined threshold"*. To help a user quickly detect the conflicts with other group members, an explanation could be formulated as follows: *"For item X, we have detected that your preference differs from the preferences of users A and B"*. Besides, explanations concerning repair actions will be proposed to give users some hints of how to adapt their preferences. For instance, *"we have found that your close friends Susan and Henry are interested in item X, and you also do not have any problem with this option. To help the group quickly reach the consensus threshold, you just need to slightly increase your rating for this item to the value V"*.

Along with the explanations, visualization methods would be necessary to facilitate the consensus making process of group members. For instance, we could use a *node-link diagram* to represent the conflict/agreement between two group members. The diagram consists of *nodes* showing the name of group members and *links* representing the *conflict/agreement* level between group members, i.e., the thicker the link, the higher the conflict/agreement level between two individual group members' preferences. The consensus status of a group of friends in the tourism domain could be visualized as shown in *Figure 9.1*.

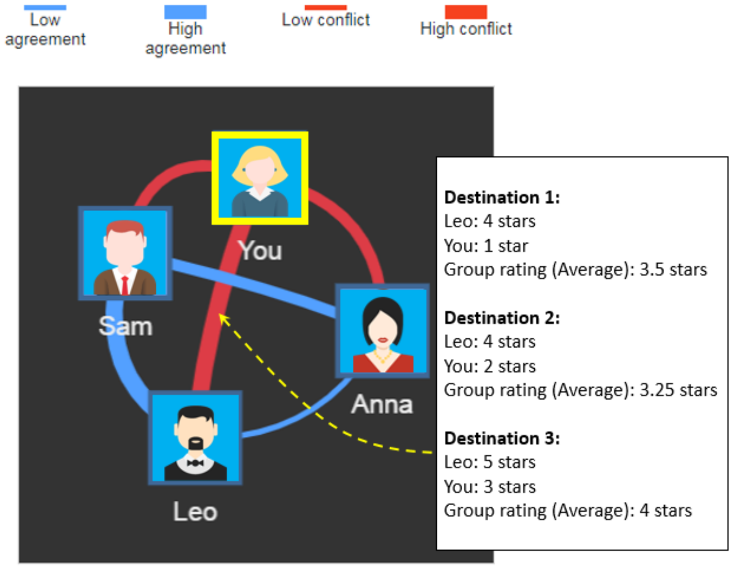


Figure 9.1.: A node-link diagram representing the conflicts/agreements among group members. A node represents a group member and a link represents the conflict/agreement between two individual group members (red link: conflict; blue link: agreement). The thickness of a link represents the gravity of a conflict or an agreement. For instance, the current user (shown by the yellow-frame picture) has a *high conflict* with Leo. By clicking on the red link, the current user can see the details of the conflicts between her preferences and Leo's.

9.2.3. Persuasive Explanations

Persuasiveness is related to the capability of an explanation to convince group members to accept group recommendations (Tintarev and Masthoff, 2007; Felfernig et al., 2018c). Recently, some studies have focused on generating group recommendations that take into account *social factors* within the group. For instance, Quijano-Sanchez et al. (2017) proposed explanations to persuade users to accept a recommended item based on the influence of social factors within the group. These explanations were created for an individual user and applied in the group decision scenarios where some group members know each other before. The social factors within a group could be the *personality* of each group member and the *closeness* among them (Quijano-Sanchez et al., 2017). Inspired by this paper, in the following paragraphs, we will present our idea to propose explanations that take into account the two mentioned social factors. The proposed explanations are assumed to increase the acceptance of users concerning group recommendations.

Personality

This factor describes the *personality* of individual group members which can be categorized into three levels: *high* (e.g., assertive), *medium* (e.g., reserved), and *low* (e.g., cooperative). An explanation regarding the personality factor can be used when a current user u (who is requesting an explanation) has a *low personality value* or a *low rating value* for item i, whereas other group members have *higher personality* *values* or *higher rating values*. The purpose of this explanation is to increase the user's acceptance of a recommended item. An example explanation could be: *"Although you do not like the item X that much, your friends Maria and Thomas like it. Besides, we have detected that they are quite assertive persons who rarely give up their decision"*. Furthermore, explanations considering other types of the current user' personality, such as *"reserved"* or *"assertive"* should also be proposed.

Closeness

This factor describes the *closeness* level between two group members, which can also be categorized into three levels: *high, medium,* and *low*. An explanation regarding the closeness is helpful in situations where the current user *u* gave a *low rating value* for the recommended item; whereas his/her close friends (whom he/she trusts) gave higher ratings for the recommended item. In this scenario, an example explanation could be: *"Although we have detected that your preference for the item X is not that high, your close friends (Maria and Thomas) whom you really trust think this option is a good choice"*. This explanation could change the idea of user *u* and encourage him/her to follow his/her close friends' opinion.

However, in the mentioned scenario, user *u* seems to be dominated by his/her (close) friends, and this could trigger his/her dissatisfaction. Therefore, the personality and closeness of group members should be taken into account together with other factors (e.g., *fairness and consensus aspects* mentioned in Chapter 5) to increase the persuasiveness of group recommendations and boost the satisfaction of users with regard to recommended items.

9.2.4. Further Group Recommendation Approaches

Most of the existing group recommendation approaches create group recommendations by combining the individual group members' preferences into a group profile using preference aggregation strategies (Masthoff, 2011; Felfernig et al., 2018a). These approaches are subject to generate group recommendations which consider as far as possible the preferences of group members and to maximize the satisfaction of group members with recommended items. In recent years, another group recommendation technique which captures different attitudes of individual group members has been paid attention by researchers (Garcia et al., 2009; Villavicencio et al., 2016). This approach utilizes a *negotiation process* to offer a recommended item. The negotiation process helps group members resolve conflicts concerning their preferences, ends up with an agreement among group members, and results in a list of constraints that matches the preferences of group members (Villavicencio et al., 2016).

For this approach, a *multi-agent system* is implemented in which each agent represents the preference of a group member. An agent communicates with other agents via a *negotiation protocol* to look for an agreement on a proposed item. The agent can use one of the three negotiation levels to express its reaction to the proposed item. In *level 1 (self-interest negotiation)*, the user negotiates with other group members taking into account his/her own constraints for items. In *level 2 (collaborative negotiation)*, the user performs the negotiation considering the constraints of other group members that were rejected before. In *level 3 (highly collaborative negotiation)*, the user performs the negotiation not only considering others' constraints but also adapting his/her initial constraints for items to reach a consensus with other group members.

Compared to approaches that apply ranking aggregation strategies, the mentioned approach achieves higher satisfaction of users about group recommendations (Villavicencio et al., 2016). However, this approach has been only tested with small-size groups (e.g., two-user groups (Villavicencio et al., 2016)). Besides, it has not taken into account the group dynamics in the negotiation process. Therefore, in the future, we will extend this approach by applying it to larger groups. In a large group, the negotiation process among group members would be much more complex. Consequently, innovative techniques should be integrated to better support the negotiation process of group members. Furthermore, we will investigate the influence of *group dynamics* (Toseland et al., 2004) on the negotiation behavior of group members. The negotiation behavior of group members might be different depending on group composition factors, such as *gender, age,* or *cultural background*. For instance, users of *homogeneous groups* (i.e., with strong cohesiveness) are assumed to use *collaborative negotiations*, whereas users of *heterogeneous groups* (i.e., with weak cohesiveness) might use *self-interest negotiations* which only take into account their preferences rather than of the preferences of other group members.

List of Figures

2.1. CHOICLA group decision support environment. Each entry represents a group decision task - the corresponding percentages indicate the share of users who already articulated their requirements. A red circle indicates the fact that the current user did not articulate his/her preferences. 21

4.1. Graphical representation of the explanation data contained in Table 4.1. 39
4.2. Graphical representation of the explanation data contained in Table 4.2. 39
4.3. Spider diagram for explaining *aggregated models* based collaborative filtering recommendations: ratings of nearest neighbor groups gp_1,.., gp_4 of gp for the recommended item t_4. This representation is a variant of consensus-based interfaces discussed in (Mahyar et al., 2017). 39
4.4. Tag-cloud representation used to show the relevance of tags with regard to a specific item extended with preference information related to group members (Isa, Joe, and Leo). 42
4.5. Visualization of the importance of interest dimensions with regard to the overall item evaluation (the importance values are based on Table 4.6 where dim_1 = *risk*, dim_2 = *profit*, and dim_3 = *availability*). 46
4.6. Visualizing the degree of fairness (Formula 4.6) in repeated group decisions (e.g., decisions on restaurant visits). In this example, the visualization indicates that user u_1 was disadvantaged in previous decisions. 46

5.1. A taxonomy of social choice-based explanations. 52

6.1. The visualization of $UI1_{GR}$ (transparency level = 2). Each row indicates that *“a group member has adapted the rating of an item”*. For instance, the first row shows that: *“Alex has adapted the rating of the mountain option”*. 74
6.2. The visualization of $UI1_{GR}$ (transparency level = 3). The blank cell indicates that a group member has not adapted the rating of an item. The *edit icon* indicates that a group member has adapted the rating of an item. 74
6.3. The $UI1_{R+TD}$ (transparency level = 4) visualizing the dimensions of *“group member”*, *“item”*, *“rating”*, and *“tendency”*. 75
6.4. The visualization of $UI1_{TL+TD+GR}$ (transparency level = 5). The blank cell indicates that a group member has not adapted the rating of an item. Signs “+” or “-” represents that a group member has increased or decreased the rating of an item. 76
6.5. The visualization of $UI2_{all}$ (transparency level = 5). Each number indicates the group rating of an item at a specific time. For instance, on 21.Jun.18, at 8:02:11, the group rating of the *“mountain”* option was 3.5 stars. 76

6.6. The $UI1_{all}$ (transparency level = 6) visualizing all dimensions concerning the rating adaptations of group members. 77

7.1. Predicting the rating value for a target recipe using a CB algorithm proposed by Freyne and Berkovsky (2010) 92

7.2. An example of recipe graph G for menu planning (Kuo et al., 2012). *"Tomato, flour, basil"* (ingredients shown with black borders) are query ingredients. The recommended menu plan is a set of recipes {*"Mozzarella, Tomato, and Basil Salad", "Lasagna", "Italian Bread*"} (nodes shown with black frames) which the total menu cost is minimal (i.e., 0.23). 93

7.3. Screenshots of the CHOICLA group decision support environment (iOS version). Figure 7.3(a) shows a list of different group decisions created by users. Users can rate alternatives by using the user interface shown in Figure 7.3(b). The suggestion for the whole group is shown in the *"Suggestion"* tab (Figure 7.3(c)). The alternative enclosed with the *medal icon* is the suggested alternative for the whole group. For instance, *"Turkish menu"* is chosen by CHOICLA to recommend to the whole group. 102

9.1. A node-link diagram representing the conflicts/agreements among group members. A node represents a group member and a link represents the conflict/agreement between two individual group members (red link: conflict; blue link: agreement). The thickness of a link represents the gravity of a conflict or an agreement. For instance, the current user (shown by the yellow-frame picture) has a *high conflict* with Leo. By clicking on the red link, the current user can see the details of the conflicts between her preferences and Leo's. 122

List of Tables

1.1. Thesis contributions corresponding to the research questions. 12

2.1. Tabular representation of constraints in a group-based configuration task (*Example 3*). Conflict set $CS_1 = \{pref_{11}, pref_{12}\}$ reflects inconsistent preferences of stakeholder 1 (the preferences are inconsistent with the configuration knowledge base) and conflict set $CS_2 = \{pref_{13}, pref_{21}\}$ reflects a conflict between the preferences of stakeholders 1 and 2. 18
2.2. Overview of the impact of the different diagnoses Δ_i on the current preferences of stakeholders, for example, stakeholder 1 has to change two of his/her requirements if diagnosis Δ_1 gets selected. 19
2.3. Evaluation of the different diagnoses using the *least misery*, *average*, and the *most pleasure* heuristics. In all three heuristics, the ranking criteria for the diagnoses is *"less is better"*. . 19

3.1. Example requirements for the development of a *sport watch*. Each requirement is described by an *id*, a *title*, and a *textual description*. 25
3.2. Defined releases for the development of a *sport watch*. Each release is described by the *start date*, the *end date,* and the *capacity*. The *capacity* indicates the planned effort of a release which is measured in hours. 26
3.3. Evaluations of stakeholders for the defined requirements in Table 3.1. Each requirement is represented by the following three dimensions: *risk, profit,* and *effort*. Each evaluation is in the range of 1 to 5. An evaluation of 5 indicates a requirement with *low risk, high profit,* and *low effort*. An evaluation of 1 represents a requirement with *high risk, low profit,* and *high effort*. Evaluations which were not provided by stakeholders are represented by dash symbols ('-'). 27
3.4. The expertise level of the experts in the *sport watch* domain. The expertise level is in the range of 1 to 5, whereby 1 indicates limited knowledge and 5 indicates excellent knowledge. 27
3.5. Experts chosen by stakeholders regarding different requirements. The dash symbol '-' represents a situation in which a stakeholder does not need any advice of an expert. 28
3.6. The evaluations of stakeholders for the defined requirements in Table 3.1. Each requirement is represented by the three following properties: *risk, profit,* and *effort*. The evaluation is in the range of 1 to 5 (5 indicates a requirement with *low risk, high profit,* and *low effort* and 1 represents a requirement with *high risk, low profit,* and *high effort*). Evaluations which were not provided by stakeholders or experts are represented by dash symbols ('-'). Evaluations provided by experts are represented in bold numbers. 29
3.7. The assignment of the requirements to releases based on the *effort* of requirements, *dependencies* between requirements, their *utility values*, and *the capacity of releases*. The effort of each requirement is represented in the second column of the table. 31

4.1. Collaborative filtering explanations for *aggregated predictions*. The explanations are based on information about the preferences (ratings) of nearest neighbors (n_{ij}) of individual group members u_i. 38
4.2. Collaborative filtering explanations for *aggregated models*, i.e., explanations are based on the aggregated preferences of individual group members. 38
4.3. Content-based filtering explanations for *aggregated predictions*. The most explanation-relevant categories for an item t_k are marked with $\surd$. 40
4.4. Opinion mining based explanations for *aggregated models*. Feature f_i with the highest explanation-relevance are marked with $\surd$. 41
4.5. Explanation relevance of requirements in constraint-based recommendation (*aggregated models*). The most relevant requirement is marked with $\surd$. 42
4.6. Explanation relevance of interest dimensions in utility-based recommendation (*aggregated predictions*). The most relevant dimension is marked with $\surd$. 43
4.7. An example of an adaptation of individual users' weights to take fairness into account. In this example, the importance (*imp*) (or the weights) of user u_1 have been increased, the weights of u_2 remain the same, and the weights of user u_3 have been decreased (the preferences of u_3 have been favored in previous decisions - a visualization is given in Figure 4.6). 45
4.8. Critiques of group members as a basis for generating explanations for item recommendations. *Support* is defined by the share of *attribute-specific critiques* supported by an item t_i. 47
4.9. Summary of the support-degree of user-specific critiques on item t_1. 48

5.1. *Type 1* - Explanations based on social choice-based preference aggregation strategies. . . . 53
5.2. *Type 2* - Explanations based on social choice-based preference aggregation strategies and decision history. 54
5.3. *Type 3* - Explanations based on social choice-based preference aggregation strategies and future decision plan. 55
5.4. Ratings of group members for the restaurants (1: *the worst*, 5: *the best*). 57
5.5. *p values* (*2-tailed*) of the Kruskal-Wallis tests in the *perceived fairness, perceived consensus,* and *satisfaction levels* across different explanations in *Type 1, Type 2,* and *Type 3*. . . . 59
5.6. *p values* (*2-tailed*) of the Mann-Whitney U tests ($\alpha' = .005$) between the *MPL-based* explanation and one of the remaining explanations in *Type 1*. 59
5.7. *p values* (*2-tailed*) of the Mann-Whitney U tests ($\alpha' = .003$) between the *MPL-based* explanation and one of the remaining explanations in *Type 3*. 59
5.8. *p values* (*2-tailed*) of the Mann-Whitney U tests ($\alpha' = .003$) between the *FAI-based* explanation and one of the remaining explanations in *Type 3*. 59
5.9. *Mean ranks* generated in the Kruskal-Wallis tests for all explanations in *Type 1* and *Type 3*. 60
5.10. *p values* (*2-tailed*) achieved from the Mann-Whitney U tests between pairs of *ADD-based, APP-based*, and *MAJ-based* explanations in *Type 1* and *Type 3*. 60
5.11. Spearman Rank Order Correlations between the *perceived fairness/perceived consensus levels* and *satisfaction levels* of all explanations in *Type 1, Type 2,* and *Type 3*. 62
5.12. Mann-Whitney U tests ($\alpha' = .025$) for all explanations between *Type 1* and *Type 2*. 63
5.13. Mann-Whitney U tests ($\alpha' = .025$) for all explanations between *Type 1* and *Type 3*. 63

6.1. The construction of the UIs of *Group 1*, where *GM - Group member, I - Item, R - Rating, TL - Timeline, TD - Tendency,* and *GR - Group recommendation*. The ratings represented in the UIs of *Group 1* are *original ratings* and *adapted ratings*. 71

6.2. The construction of the UIs of *Group 2* where *I - Item, TL - Timeline, R - Rating, TD - Tendency,* and *GR - Group recommendation*. The ratings represented in the UIs of *Group 2* are *group ratings*. 71
6.3. The UIs of *Group 1* and *Group 2* which are selected for the user study. 72
6.4. Ratings of group members for the destination types (1 = *the worst*, 5 = *the best*). 73
6.5. The distribution of the participants for the UIs at different transparency levels. 78
6.6. Kruskal-Wallis test ($\alpha = .05, p = .000$) across different transparency levels of the UIs in *Group 1*. The higher the mean rank, the lower the preparedness level of decision manipulation. 80
6.7. Kruskal-Wallis test for all UIs at the *transparency level of 3* ($\alpha = .05, p = .009$). The higher the mean rank, the lower the preparedness level of decision manipulation. 81
6.8. Kruskal-Wallis test for all UIs at the *transparency level of 4* ($\alpha = .05, p = .018$). The higher the mean rank, the lower the preparedness level of decision manipulation. 81
6.9. Kruskal-Wallis test for all UIs at the *transparency level of 5* ($\alpha = .05, p = .038$). The higher the mean rank, the lower the preparedness level of decision manipulation. 82
6.10. Kruskal-Wallis test ($\alpha = .05, p = .008$) in the understandability levels across different UIs. 83
6.11. Kruskal-Wallis test in the influence levels across different dimensions of the $UI1_{TL+TD+GR}$ ($\alpha = .05, p = .024$). 83
6.12. Kruskal-Wallis test in the influence levels across different dimensions of the $UI1_{all}$ ($\alpha = .05, p = .006$). 83

7.1. A summary of state-of-the-art of recommender systems in the healthy food domain (CF: Collaborative filtering recommender systems, CB: Content-based recommender systems) . 87
7.2. A list of available menus with corresponding information 94
7.3. Reference values for nutritional intake. Bonn. 2. Edition, 1. Volume (2015) published by German Nutrition Association, Austrian Nutrition Association, and Swiss Nutrition Association. 95
7.4. A list of recommended menus to user u_a 95
7.5. The constraint satisfaction problem modeled with a *parameter-based approach* (Aberg, 2006). 97
7.6. A list of menus recommended to user u_a on the basis of considering his favorite ingredients (i.e., *tomato*) and nutritional needs. 98
7.7. An example of using *Least Misery* strategy to aggregate individual user preferences into a group profile. $recipe_1$ is recommended to the group since its group rating value is highest. 101
7.8. A summary of research challenges in food recommender systems and proposed solutions. . 103

8.1. A predefined group decision making scenario where group members explicitly evaluate different holiday destinations using a 5-star rating scale (1: the worst, 5: the best). 109
8.2. On the basis of the group members' evaluations for the holiday destinations (see *Table* 8.1), group aggregation strategies are used to recommend a corresponding holiday to the group. 109
8.3. Settings of user preferences (evaluations) used in the user study (S_1 - S_5). Preferences are expressed in terms of ratings on a 5-star rating scale (1: the worst, 5: the best) 110
8.4. Tasks used in the user study. The settings were taken from Table 8.3. $M = \binom{5}{3} = 10$ tasks represent all possible combinations of three out of five settings. 111
8.5. Possible sequences of four chosen domains. For instance, H-A-M-R denotes a sequence with the following order: *1st*: H (holiday), *2nd*: A (shared-apartment), *3rd*: M (music genre), and *4th*: R (restaurant). 112
8.6. The distribution of the average time consumption of the participants for *Task 4* in different item domains. 113

8.7. An analysis of the precision (i.e., the prediction quality) of group aggregation strategies in two types of domain sequences of *Task 4*. *Task 4* represents a scenario in which, for most items, the preferences of a minority of group members differ from those of other group members (see Table 8.1). 114

Bibliography

B. Abdollahi and O. Nasraoui. 2017. Using Explainability for Constrained Matrix Factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 79–83. (Cited on page 48.)

J. Aberg. 2006. Dealing with Malnutrition: A Meal Planning System for Elderly. In *AAAI Spring Symposium: Argumentation for Consumers of Healthcare*. AAAI, 1–7. (Cited on pages 1, 7, 87, 96, 97, 102, and 129.)

G. Adomavicius, J. Bockstedt, S. Curley, and J. Zhang. 2011. Recommender Systems, Consumer Preferences, and Anchoring Effects. In *RecSys 2011 - Workshop on Human Decision Making in Recommender Systems*, Vol. 811. 35–42. (Cited on page 90.)

K. Aizawa, G. C. de Silva, M. Ogawa, and Y. Sato. 2010. Food Log by Snapping and Processing Images. In *The 16th International Conference on Virtual Systems and Multimedia*. IEEE, 71–74. (Cited on page 103.)

S. Alonso, S. Herrera-Viedma, F. Chiclana, and F. Herrera. 2010. A Web Based Consensus Support System for Group Decision Making Problems and Incomplete Preferences Information Sciences. *Information Sciences* 180, 30 (Dec. 2010), 4477–4495. (Cited on page 121.)

S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu. 2009. Group Recommendation: Semantics and Efficiency. *Proc. VLDB Endow.* 2, 1 (Aug. 2009), 754–765. (Cited on page 44.)

L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. 2003. Intrigue: Personalized Recommendation of Tourist Attractions for Desktop and Hand Held Devices. *Applied Artificial Intelligence* 17, 8-9 (2003), 687–714. (Cited on pages 2, 36, 43, 51, 86, and 89.)

D. Asanov. 2011. Algorithms and Methods in Recommender Systems. Berlin Institute of Technology, Berlin, Germany, 1–7. (Cited on page 88.)

M. Atas, A. Felfernig, M. Stettinger, and T. N. T. Tran. 2017. Beyond Item Recommendation: Using Recommendations to Stimulate Knowledge Sharing in Group Decisions. In *Social Informatics - 9th International Conference, SocInfo 2017, Proceedings (Lecture Notes in Computer Science)*, Vol. 10540 LNCS. Springer Verlag Wien, 368–377. (Cited on pages 35 and 48.)

M. Atas, T. N. T. Tran, R. Samer, A. Felfernig, M. Stettinger, and D. Fucci. 2018. Liquid Democracy in Group-based Configuration. In *Proceedings of the 20th Configuration Workshop, Graz, Austria, September 27-28, 2018.* 93–98. (Cited on pages 9 and 23.)

M. Balabanović and Y. Shoham. 1997. Fab: Content-based, Collaborative Recommendation. *Communication ACM* 40, 3 (March 1997), 66–72. (Cited on page 88.)

D. J. Beal, R. R. Cohen, M. Burke, and C. L. McLendon. 2003. Cohesion and Performance in Groups: A Meta-analytic Clarification of Construct Relations. *The Journal of applied psychology* 88, 6 (2003), 989–1004. (Cited on page 119.)

S. Berkovsky and J. Freyne. 2010. Group-based Recipe Recommendations: Analysis of Data Aggregation Strategies. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 111–118. (Cited on pages 7, 87, 89, 99, 100, 103, and 105.)

M. Bilgic and R. J. Mooney. 2005. Explaining Recommendations: Satisfaction vs. Promotion. In *Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces*. San Diego, CA, 1–6. (Cited on pages 33, 34, 37, 40, and 116.)

C. Blum and C. I. Zuber. 2016. Liquid Democracy: Potentials, Problems, and Perspectives. *The Journal of Political Philosophy* 24 (2016), 162–182. (Cited on pages 26 and 116.)

D. K. Bokde, S. Girase, and D. Mukhopadhyay. 2015. Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey. *Procedia Computer Science* 49 (Apr. 2015), 136–146. (Cited on page 88.)

P. Boldi, C. Monti, M. Santini, and S. Vigna. 2015. Liquid FM: Recommending Music through Viscous Democracy. In *Proceedings of the 6th Italian Information Retrieval Workshop, Cagliari, Italy, May 25-26, 2015.* 1–12. (Cited on pages 24 and 116.)

V. L. Borgan, D. R. Just, and C. S. Dev. 2012. Team Gender Diversity and Investment Decision Making behavior. *Review of Behavioral Finance* 5, 2 (2012), 134–152. (Cited on page 120.)

J. Borrís, A. Moreno, and A. Valls. 2014. Intelligent Tourism Recommender Systems: A Survey. *Expert Syst. Appl.* 41, 16 (Nov. 2014), 7370–7389. (Cited on page 1.)

D. Bridge, M. H. Göker, L. McGinty, and B. Smyth. 2005. Case-based Recommender Systems. *Knowl. Eng. Review* 20, 3 (Sept. 2005), 315–320. (Cited on page 88.)

B. Buchanan and E. Shortliffe. 1985. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley. (Cited on page 41.)

R. Burke. 2000. Knowledge-based Recommender Systems. In *Encyclopedia of Library and Information Systems*. Vol. 69. Marcel Dekker, 180–200. (Cited on pages 1, 86, and 88.)

R. Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12, 4 (Nov. 2002), 331–370. (Cited on page 89.)

R. Burke, A. Felfernig, and M. H. Göker. 2011. Recommender Systems: An Overview. *AI Magazine* 32 (2011), 13–18. (Cited on pages 1 and 86.)

I. Cantador and P. Castells. 2012. Group Recommender Systems: New Perspectives in the Social Web. In *Recommender Systems for the Social Web*. Intelligent Systems Reference Library, Vol. 32. Springer Berlin Heidelberg, 139–157. (Cited on pages 89 and 101.)

G. Carenini and J. D. Moore. 2006. Generating and Evaluating Evaluative Arguments. *Artificial Intelligence* 170, 11 (2006), 925 – 952. (Cited on page 43.)

J. Castro, J. Lu, G. Zhang, Y. Y. Dong, and L. Martnez. 2018. Opinion Dynamics-Based Group Recommender Systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48, 12 (Dec. 2018), 2394–2406. (Cited on page 44.)

J. Castro, F. J. Quesada, I. Palomares, and L. Martínez. 2015. A Consensus-Driven Group Recommender System. *Int. J. Intell. Syst.* 30, 8 (Aug. 2015), 887–906. (Cited on pages 44 and 106.)

S. Chang, F. Maxwell Harper, L. He, and L. G. Terveen. 2016. CrowdLens: Experimenting with Crowd-Powered Recommendation and Explanation. In *ICWSM*. AAAI Press, 52–61. (Cited on page 36.)

L. Chen, M. de Gemmis, A. Felfernig, P. Lops, F. Ricci, and G. Semeraro. 2013. Human Decision Making and Recommender Systems. *ACM Trans. Interact. Intell. Syst.* 3, 3, Article 17 (Oct. 2013), 7 pages. (Cited on page 49.)

L. Chen and P. Pu. 2012. Critiquing-based Recommenders: Survey and Emerging Trends. *User Modeling and User-Adapted Interaction* 22, 1 (01 Apr 2012), 125–150. (Cited on pages 34, 35, 47, 50, and 101.)

Y. Chen. 2011. Interface and Interaction Design for Group and Social Recommender Systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*. ACM, New York, NY, USA, 363–366. (Cited on page 36.)

F. Chiclana, F. Mata, S. Alonso, E. Herrera-Viedma, and L. Martínez. 2007. Group Decision Making: From Consistency to Consensus. In *Modeling Decisions for Artificial Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg, 80–91. (Cited on pages 2 and 50.)

F. Chiclana, J. Wu, and E. Herrera-Viedma. 2014. *Visual Consensus Feedback Mechanism for Group Decision Making with Complementary Linguistic Preference Relations*. Springer International Publishing, Cham, 72–83. (Cited on page 121.)

P. A. Chirita, W. Nejdl, and C. Zamfir. 2005. Preventing Shilling Attacks in Online Recommender Systems. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management (WIDM '05)*. ACM, New York, NY, USA, 67–74. (Cited on pages 6, 67, and 68.)

I. K. Chung and C. R. Adams. 1997. A Study on the Characteristics of Group Decision Making Behavior: Cultural Difference Perspective of Korea vs. U.S. *J. Glob. Inf. Manage.* 5, 3 (1997), 18–29. (Cited on pages 107 and 119.)

J. Cohen. 1988. *Statistical power analysis for the behavioral sciences*. Routledge. (Cited on page 61.)

V. Conitzer and T. Sandholm. 2002. Complexity of Mechanism Design. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI'02)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 103–110. (Cited on page 69.)

V. Conitzer and M. Yokoo. 2010. Using Mechanism Design to Prevent False-Name Manipulations. *AI Magazine* 31, 4 (2010), 65–78. (Cited on pages 4, 6, 7, 68, and 69.)

R. Coster, A. Gustavsson, T. Olsson, A. Rudström, and A. Rudström. 2002. Enhancing Web-Based Configuration With Recommendations And Cluster-Based Help. In *In Proceedings of the AH'2002 Workshop on Recommendation and Personalization in eCommerce*. 11 pages. (Cited on page 3.)

H. Cramer, V. Evers, S. Ramlal, M. Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga. 2008. The Effects of Transparency on Trust in and Acceptance of a Content-based Art Recommender. *User Modeling and User-Adapted Interaction* 18, 5 (Nov. 2008), 455–496. (Cited on pages 33 and 40.)

A. Crossen, J. Budzik, and K. J. Hammond. 2002. Flytrap: Intelligent Group Music Recommendation. In *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI'02)*. ACM, New York, NY, USA, 184–185. (Cited on page 86.)

G. Desanctis and R. B. Gallupe. 1987. A Foundation for the Study of Group Decision Support Systems. *Manage. Sci.* 33, 5 (May 1987), 589–609. (Cited on page 119.)

R. Dong, M. Schaal, M. P. O'Mahony, and B. Smyth. 2013. Topic Extraction from Online Reviews for Classification and Recommendation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI '13)*. AAAI Press, 1310–1316. (Cited on page 41.)

J. S. Dyer. 2005. *MAUT - Multi-attribute Utility Theory*. Springer New York, New York, NY, 265–292. (Cited on pages 9, 24, 28, and 116.)

A. Eagly and B. Johnson. 1990. Gender and Leadership Style: A Meta-analysis. *Psychological Bulletin* 108, 2 (1990), 233–256. (Cited on page 120.)

M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. 2011. Collaborative Filtering Recommender Systems. *Found. Trends Hum.-Comput. Interact.* 4, 2 (Feb. 2011), 81–173. (Cited on pages 67 and 88.)

M. A. El-Dosuky, M. Z. Rashad, T. T. Hamza, and A. H. El-Bassiouny. 2012. Food Recommendation Using Ontology and Heuristics. In *AMLTA (Communications in Computer and Information Science)*, Vol. 322. Springer, 423–429. (Cited on pages 7, 87, 91, and 117.)

M. Elahi, M. Ge, F. Ricci, I. Fernndez-Tobas, S. Berkovsky, and D. Massimo. 2015. Interaction Design in a Mobile Food Recommender System. In *IntRS@RecSys (CEUR Workshop Proceedings)*, Vol. 1438. CEUR-WS.org, 49–52. (Cited on pages 7, 87, 91, and 92.)

M. Elahi, M. Ge, F. Ricci, D. Massimo, and S. Berkovsky. 2014. Interactive Food Recommendation for Groups. In *RecSys'14*, Vol. 1247. 1–2. (Cited on pages 87, 99, 101, 103, 105, and 117.)

D. Elsweiler, M. Harvey, B. Ludwig, and A. Said. 2015. Bringing the "Healthy" into Food Recommenders. In *DMRS (CEUR Workshop Proceedings)*, Vol. 1533. CEUR-WS.org, 33–36. (Cited on pages 7, 87, 90, 98, 99, and 117.)

S. P. Erdeniz, I. Maglogiannis, A. Menychtas, A. Felfernig, and T. N. T. Tran. 2018. Recommender Systems for IoT Enabled m-Health Applications. In *Artificial Intelligence Applications and Innovations*. Springer International Publishing, Cham, 227–237. (Cited on page 7.)

A. Falkner, A. Felfernig, and A. Haag. 2011. Recommendation Technologies for Configurable Products. *AI Magazine* 32, 3 (2011), 99–108. (Cited on page 3.)

A. Felfernig. 2014. Biases in Decision Making. In *Proceedings of the First International Workshop on Decision Making and Recommender Systems (DMRS2014), Bolzano, Italy, September 18-19, 2014.* Vol. 1278. CEUR Proceedings, 32–37. (Cited on pages 90 and 102.)

A. Felfernig. 2016. Application of Constraint-based Technologies in Financial Services Recommendation. In *2nd International Workshop on Personalization and Recommender Systems in Financial Services (FinRec'16)*. 22–23. (Cited on page 1.)

A. Felfernig, M. Atas, D. Helic, T. N. T. Tran, M. Stettinger, and R. Samer. 2018a. *Group Recommender Systems: An Introduction*. Springer, Chapter Algorithms for Group Recommendation, 27–58. (Cited on pages 2, 9, 10, 12, 35, 36, 37, 41, 44, 51, 52, 69, 109, 116, 118, and 123.)

A. Felfernig, M. Atas, M. Stettinger, T. N. T. Tran, and G. Leitner. 2018b. *Group Recommender Systems: An Introduction*. Springer, Chapter Biases in Group Decisions, 145–155. (Cited on pages 3, 35, and 70.)

A. Felfernig, M. Atas, T. N. T. Tran, and M. Stettinger. 2016. Towards Group-based Configuration. In *Proceedings of the 18th International Configuration Workshop September 5-6, 2016, Toulouse, France (ConfWS16)*. 69–72. (Cited on pages 2, 9, 15, 23, 44, and 50.)

A. Felfernig, M. Atas, T. N. T. Tran, M. Stettinger, S. P. Erdeniz, and G. Leitner. 2017. An Analysis of Group Recommendation Heuristics for High- and Low-Involvement Items. In *IEA/AIE 2017, Arras, France*. 335–344. (Cited on pages 8, 44, 50, and 107.)

A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčič. 2018c. *Personality, Emotions, and Group Dynamics*. Springer International Publishing, Cham, 157–167. (Cited on pages 118 and 122.)

A. Felfernig and R. Burke. 2008. Constraint-based Recommender Systems: Technologies and Research Issues. In *Proceedings of the 10th International Conference on Electronic Commerce (ICEC'08)*. ACM, New York, NY, USA, Article 3, 10 pages. (Cited on pages 2, 35, and 88.)

A. Felfernig, G. Friedrich, B. Gula, M. Hitz, T. Kruggel, G. Leitner, R. Melcher, D. Riepan, S. Strauss, E. Teppan, and O. Vitouch. 2007a. Persuasive Recommendation: Serial Position Effects in Knowledge-based Recommender Systems. In *Proceedings of the 2nd International Conference on Persuasive Technology (PERSUASIVE'07)*. Springer-Verlag, Berlin, Heidelberg, 283–294. (Cited on pages 4 and 108.)

A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. 2011. *Recommender Systems Handbook*. Springer US, Chapter: Developing Constraint-based Recommenders, 187–215. (Cited on page 88.)

A. Felfernig, G. Friedrich, M. Schubert, M. Mandl, M. Mairitsch, and E. Teppan. 2009. Plausible Repairs for Inconsistent Requirements. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*. 791–796. (Cited on pages 3 and 42.)

A. Felfernig, G. E. Friedrich, and D. Jannach. 2000a. UML as Domain Specific Language for The Construction of Knowledge-Based Configuration Systems. *International Journal of Software Engineering and Knowledge Engineering (IJSEKE* 10 (2000), 449–469. (Cited on page 4.)

A. Felfernig, G. E. Friedrich, D. Jannach, and M. Stumptner. 2000b. Consistency-based Diagnosis of Configuration Knowledge Bases. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'00)*. IOS Press, Amsterdam, The Netherlands, 146–150. (Cited on page 4.)

A. Felfernig, B. Gula, G. Leitner, M. Maier, R. Melcher, S. Schippel, and E. Teppan. 2008b. A Dominance Model for the Calculation of Decoy Products in Recommendation Environments. In *In AISB Symposium on Persuasive Technologies*. 43–50. (Cited on page 34.)

A. Felfernig, B. Gula, G. Leitner, M. Maier, R. Melcher, and E. Teppan. 2008a. Persuasion in Knowledge-Based Recommendation. In *Persuasive Technology*. Springer Berlin Heidelberg, Berlin, Heidelberg, 71–82. (Cited on page 43.)

A. Felfernig, B. Gula, and E. Teppan. 2007b. Knowledge-based Recommender Technologies for Marketing and Sales. *International journal of pattern recognition and artificial intelligence (IJPRAI)* 21, 2 (2007), 333–354. I. (Cited on pages 35 and 42.)

A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen. 2014a. *Knowledge-based Configuration: From Research to Business Cases* (1 ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (Cited on pages 5, 15, 20, 23, 35, and 90.)

A. Felfernig, K. Isak, K. Szabo, and P. Zachar. 2007c. The VITA Financial Services Sales Support Environment. In *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence - Volume 2 (IAAI'07)*. AAAI Press, 1692–1699. (Cited on page 15.)

A. Felfernig, M. Schubert, and C. Zehentner. 2012a. An Efficient Diagnosis Algorithm for Inconsistent Constraint Sets. *AI EDAM* 26, 1 (2012), 53–62. (Cited on pages 4 and 44.)

A. Felfernig, M. Stettinger, G. Ninaus, S. Reiterer, M. Jeran, A. Falkner, J. Tiihonen, and G. Leitner. 2014b. Towards Open Configuration. In *Proceedings of the 16th International Workshop on Configuration*. 89–94. (Cited on pages 3, 4, 15, 16, 103, and 105.)

A. Felfernig, E. Teppan, and B. Gula. 2006. Knowledge-based Recommender Technologies for Marketing and Sales. *Pattern Recognition and Artificial Intelligence* 21, 2 (2006), 1–22. (Cited on page 88.)

A. Felfernig, N. Tintarev, T. N. T. Tran, and M. Stettinger. 2018d. *Group Recommender Systems: An Introduction*. Springer, Chapter Explanations for Groups, 105–126. (Cited on pages 6, 10, 33, 49, 50, 51, and 116.)

A. Felfernig, C. Zehentner, G. Ninaus, H. Grabner, W. Maalej, D. Pagano, L. Weninger, and F. Reinfrank. 2012b. *Advances in User Modeling*. Springer Berlin Heidelberg, Chapter: Group Decision Support for Requirements Negotiation, 105–116. (Cited on pages 15, 20, and 90.)

G. Fleischanderl, G. E. Friedrich, A. Haselböck, H. Schreiner, and M. Stumptner. 1998. Configuring Large Systems Using Generative Constraint Satisfaction. *IEEE Intelligent Systems* 13, 4 (July 1998), 59–68. (Cited on page 15.)

D. Forsyth. 2006. *Group Dynamics*. Thomson Higher Education. (Cited on page 118.)

J. Freyne and S. Berkovsky. 2010. Intelligent Food Planning: Personalized Recipe Recommendation. In *Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI '10)*. ACM, New York, NY, USA, 321–324. (Cited on pages 86, 87, 91, 92, 103, 104, 117, and 126.)

J. Freyne, S. Berkovsky, and G. Smith. 2011. Recipe Recommendation: Accuracy and Reasoning. In *19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011*. Springer Berlin Heidelberg, 99–110. (Cited on pages 7, 87, and 100.)

G. Friedrich. 2004. Elimination of Spurious Explanations. In *ECAI'2004*. 813–817. (Cited on pages 41, 42, and 43.)

G. Friedrich and M. Zanker. 2011. A Taxonomy for Generating Explanations in Recommender Systems. *AI Magazine* (2011), 90–98. (Cited on pages 33, 34, and 41.)

E. Gansner, Y. Hu, S. Kobourov, and C. Volinsky. 2009. Putting Recommendations on the Map: Visualizing Clusters and Relations. In *Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09)*. ACM, New York, NY, USA, 345–348. (Cited on page 34.)

I. Garcia, L. Sebastia, E. Onaindia, and C. Guzman. 2009. A Group Recommender System for Tourist Activities. In *E-Commerce and Web Technologies*. Springer Berlin Heidelberg, Berlin, Heidelberg, 26–37. (Cited on pages 115 and 123.)

F. Gedikli, D. Jannach, and M. Ge. 2014. How Should I Explain? A Comparison of Different Explanation Types for Recommender Systems. *Int. J. Hum.-Comput. Stud.* 72, 4 (April 2014), 367–382. (Cited on pages 3, 34, 40, 41, and 50.)

S. Gkika and G. Lekakos. 2014. The Persuasive Role of Explanations in Recommender Systems. In *Proceedings of the International Worskshop on Behavior Change Support Systems - BCSS 2014*. 59–68. (Cited on pages 6 and 35.)

D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. 1992. Using Collaborative Filtering to Weave an Information Tapestry. *Commun. ACM* 35, 12 (Dec. 1992), 61–70. (Cited on page 1.)

C. A. Gomez-Uribe and N. Hunt. 2015. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans. Manage. Inf. Syst.* 6, 4, Article 13 (Dec. 2015), 19 pages. (Cited on page 1.)

T. Greitemeyer and S. Schulz-Hardt. 2003. Preference-Consistent Evaluation of Information in the Hidden Profile Paradigm: Beyond Group-Level Explanations for the Dominance of Shared Information in Group Decisions. *Journal of personality and social psychology* 84 (03 2003), 322–339. (Cited on pages 5 and 15.)

I. Gunes, C. Kaleli, A. Bilge, and H. Polat. 2014. Shilling Attacks Against Recommender Systems: A Comprehensive Survey. *Artif. Intell. Rev.* 42, 4 (Dec. 2014), 767–799. (Cited on pages 6 and 68.)

A. Haag. 1998. Sales Configuration in Business Processes. *IEEE Intelligent Systems and their Applications* 13, 4 (Jul 1998), 78–85. (Cited on pages 5, 15, and 23.)

J. R. Hackman and N. Vidmar. 1970. Effects of Size and Task Type on Group Performance and Member Reactions. *Sociometry* 33, 1 (1970), 37–54. (Cited on page 119.)

R. J. Hannagan and C. W. Larimer. 2010. Does Gender Composition Affect Group Decision Outcomes? Evidence from a Laboratory Experiment. *Political Behavior* 32, 1 (March 2010), 51–67. (Cited on page 120.)

M. Harvey and D. Elsweiler. 2015. Automated Recommendation of Healthy, Personalised Meal Plans. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 327–328. (Cited on page 99.)

R. Hastie and T. Kameda. 2005. The Robust Beauty of Majority Rules in Group Decisions. *American Psychological Association* 112, 2 (2005), 494–508. (Cited on page 60.)

F. Hayes-Roth, D. Waterman, and D. Lenat. 1983. *Building Expert Systems*. Addison Wesley. (Cited on page 4.)

C. He, D. Parra, and K. Verbert. 2016. Interactive Recommender Systems: A Survey of the State of the art and Future Research Challenges and Opportunities. *Expert Syst. Appl.* 56, C (2016), 9–27. (Cited on pages 6 and 73.)

J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. 1999. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*. ACM, New York, NY, USA, 230–237. (Cited on page 67.)

J. L. Herlocker, J. A. Konstan, and J. Riedl. 2000. Explaining Collaborative Filtering Recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW '00)*. ACM, New York, NY, USA, 241–250. (Cited on pages 6, 33, 34, 36, 37, 38, and 50.)

J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53. (Cited on page 67.)

M. Hertzberg, R. Smith, and R. Westphal. 2013. *A Consensus Handbook: Cooperative Decision Making for Activists, Coops and Communities* (1st ed.). Seeds for Change Lancaster Co-operative Ltd. (Cited on page 50.)

J. A. Hoxmeier and C. Dicesare. 2000. System Response Time and User Satisfaction: An Experimental Study of Browser-based Applications. In *Proceedings of the Association of Information Systems Americas Conference*. 10–13. (Cited on page 104.)

A. Jameson. 2004. More Than the Sum of Its Members: Challenges for Group Recommender Systems. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 48–54. (Cited on pages 36, 44, and 117.)

A. Jameson, S. Baldes, and T. Kleinbauer. 2004. Two Methods for Enhancing Mutual Awareness in a Group Recommender System. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 447–449. (Cited on pages 15, 36, 68, 69, and 90.)

A. Jameson, C. Hackl, T. Kleinbauer, and H. Presented. 2003. Evaluation of Automatically Designed Mechanisms. , 11 pages. (Cited on pages 4, 6, and 68.)

A. Jameson and B. Smyth. 2007. The Adaptive Web. Springer-Verlag, Berlin, Heidelberg, Chapter Recommendation to Groups, 596–627. (Cited on pages 2, 36, 43, 48, 69, and 89.)

A. Jameson, M. C. Willemsen, A. Felfernig, M. D. Gemmis, P. Lops, G. Semeraro, and L. Chen. 2015. *Human Decision Making and Recommender systems* (2nd ed.). Springer, Germany, 611–648. (Cited on page 34.)

D. Jannach, S. Naveed, and M. Jugovac. 2017. User Control in Recommender Systems: Overview and Interaction Challenges. In *E-Commerce and Web Technologies*. Springer International Publishing, Cham, 21–33. (Cited on pages 4, 6, and 68.)

D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. 2010. *Recommender Systems: An Introduction* (1st ed.). Cambridge University Press, New York, NY, USA. (Cited on pages 3, 5, 16, 23, and 49.)

G. Jawaheer, P. Weller, and P. Kostkova. 2014. Modeling User Preferences in Recommender Systems: A Classification Framework for Explicit and Implicit User Feedback. *ACM Trans. Interact. Intell. Syst.* 4, 2, Article 8 (June 2014), 26 pages. (Cited on page 67.)

T. Johann and W. Maalej. 2015. Democratic Mass Participation of Users in Requirements Engineering?. In *23rd IEEE International Requirements Engineering Conference, RE'2015, Ottawa, ON, Canada, August 24-28, 2015*. 256–261. (Cited on page 24.)

U. Junker. 2004. QUICKXPLAIN: Preferred Explanations and Relaxations for Over-constrained Problems. In *Proceedings of the 19th National Conference on Artifical Intelligence (AAAI'04)*. AAAI Press, 167–172. (Cited on pages 5 and 17.)

J. Kacprzyk and S. Zadrożny. 2016. *Towards a Fairness-Oriented Approach to Consensus Reaching Support Under Fuzzy Preferences and a Fuzzy Majority via Linguistic Summaries.* Springer Berlin Heidelberg, Berlin, Heidelberg, 189–211. (Cited on pages 2 and 50.)

D. Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux. (Cited on page 3.)

O. Kapcak, S. Spagnoli, V. Robbemond, V. Vadali, S. Najafian, and N. Tintarev. 2018. TourExplain: A Crowdsourcing Pipeline for Generating Explanations for Groups of Tourists. In *Proceedings of the Workshop on Recommenders in Tourism, RecTour 2018, co-located with the 12th ACM Conference on Recommender Systems (RecSys 2018), Vancouver, Canada, October 7, 2018*. 33–36. (Cited on pages 2, 6, and 51.)

S. Kapoor. 2017. A Review of Attacks and Its Detection Attributes on Collaborative Recommender Systems. *International Journal of Advanced Research in Computer Science* 8, 7 (July 2017), 1188–1193. (Cited on page 1.)

L. Kathlene. 1994. Power and Influence in State Legislative Policymaking: The Interaction of Gender and Position in Committee Hearing Debates. *The American Political Science Review* 88 (1994), 560–576. (Cited on page 120.)

R. T. Khasawneh and E. A. Abu-Shanab. 2013. Factors Influencing Group Decision Making Performance in a GSS Enabled Environment. *Computer Science and Information Technology* 1 (2013), 145–152. (Cited on pages 107 and 120.)

W. C. Knowler, E. Barrett-Connor, S. E. Fowler, R. F. Hamman, J. M. Lachin, E. A. Walker, and D. M. Nathan. 2002. Reduction in the Incidence of Type 2 Diabetes with Lifestyle Intervention or Metformin. *N Engl J Med* 346, 6 (Feb 2002), 393–403. (Cited on page 90.)

E. Knutov, P. De Bra, and M. Pechenizkiy. 2009. AH 12 Years Later: A Comprehensive Survey of Adaptive Hypermedia Methods and Techniques. *New Review of Hypermedia and Multimedia* 15, 1 (2009), 5–38. (Cited on page 41.)

J. A. Konstan and J. Riedl. 2012. Recommender Systems: From Algorithms to User Experience. *User Modeling and User-Adapted Interaction* 22, 1 (01 Apr 2012), 101–123. (Cited on page 48.)

Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37. (Cited on page 88.)

P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, and L. Getoor. 2017. User Preferences for Hybrid Explanations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 84–88. (Cited on pages 36 and 48.)

F. F. Kuo, C. T. Li, M. K. Shan, and S. Y. Lee. 2012. Intelligent Menu Planning: Recommending Set of Recipes by Ingredients. In *Proceedings of the ACM Multimedia 2012 Workshop on Multimedia for Cooking and Eating Activities (CEA '12)*. ACM, New York, NY, USA, 1–6. (Cited on pages 87, 91, 93, 117, and 126.)

B. Lamche, U. Adigüzel, and W. Wörndl. 2014. Interactive Explanations in Mobile Shopping Recommender Systems. In *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS 14)* (2014). 14–21. (Cited on pages 6 and 34.)

J. Lang, M. Spear, and S. F. Wu. 2010. Social Manipulation of Online Recommender Systems. In *Proceedings of the Second International Conference on Social Informatics (SocInfo'10)*. Springer-Verlag, Berlin, Heidelberg, 125–139. (Cited on page 68.)

K. Lang. 1995. NewsWeeder: Learning to Filter Netnews. In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*. Morgan Kaufmann, 331–339. (Cited on page 88.)

C. Li and Z. Luo. 2011. Detection of Shilling Attacks in Collaborative Filtering Recommender Systems. In *Proceedings of the International Conference of Soft Computing and Pattern Recognition (SoCPaR) (SoCPaR'11)*. 190–193. (Cited on pages 4, 6, and 68.)

Henry Lieberman, Neil W. Van Dyke, and Adrian S. Vivacqua. 1999a. Let's Browse: A Collaborative Web Browsing Agent. In *Proceedings of the 4th International Conference on Intelligent User Interfaces (IUI '99)*. ACM, New York, NY, USA, 65–68. (Cited on pages 40 and 86.)

H. Lieberman, N. W. Van Dyke, and A. S. Vivacqua. 1999b. Let's Browse: A Collaborative Web Browsing Agent. In *Proceedings of the 4th International Conference on Intelligent User Interfaces (IUI '99)*. ACM, New York, NY, USA, 65–68. (Cited on page 67.)

W. Lin, S. A. Alvarez, and C. Ruiz. 2002. Efficient Adaptive-Support Association Rule Mining for Recommender Systems. *Data Min. Knowl. Discov.* 6, 1 (Jan. 2002), 83–105. (Cited on page 36.)

G. Linden, B. Smith, and J. York. 2003. Amazon.com Recommendations: Item-to-item Collaborative Filtering. *IEEE Internet Computing* 7, 1 (Jan. 2003), 76–80. (Cited on page 1.)

A. Litvinenko. 2012. Social Media and Perspectives of Liquid Democracy on the Example of Political Communication of Pirate Party in Germany. In *Proceedings of the 12th European Conference on e-Government in Barcelona*. 403–408. (Cited on page 26.)

P. Lops, M. D. Gemmis, and G. Semeraro. 2011. *Content-based Recommender Systems: State of the Art and Trends*. Springer US, Boston, MA, 73–105. (Cited on page 1.)

F. Lu and N. Tintarev. 2018. A Diversity Adjusting Strategy with Personality for Music Recommendation. In *Proceedings of the 5th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, IntRS 2018, co-located with ACM Conference on Recommender Systems (RecSys 2018), Vancouver, Canada, October 7, 2018.* 7–14. (Cited on page 1.)

N. Mahyar, W. Liu, S. Xiao, J. Browne, M. Yang, and S. P. Dow. 2017. ConsesnsUs: Visualizing Points of Disagreement for Multi-Criteria Collaborative Decision Making. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17 Companion)*. ACM, New York, NY, USA, 17–20. (Cited on pages 38, 39, 44, and 125.)

M. Mandl, A. Felfernig, E. Teppan, and M. Schubert. 2011. Consumer Decision Making in Knowledge-based Recommendation. *Journal of Intelligent Information Systems* 37, 1 (01 Aug 2011), 1–22. (Cited on pages 3 and 118.)

J. Mankoff, G. Hsieh, H. C. Hung, S. Lee, and E. Nitao. 2002. Using Low-Cost Sensing to Support Nutritional Awareness. In *Ubicomp (Lecture Notes in Computer Science)*, Vol. 2498. Springer, 371–376. (Cited on page 105.)

J. Masthoff. 2004. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *User Modeling and User-Adapted Interaction* 14, 1 (Feb 2004), 37–85. (Cited on pages 16, 89, 107, and 108.)

J. Masthoff. 2011. *Group Recommender Systems: Combining Individual Models*. Springer, 677–702. (Cited on pages 2, 8, 9, 10, 12, 16, 19, 35, 44, 51, 52, 70, 89, 90, 100, 101, 102, 105, 107, 108, 112, 116, 118, and 123.)

J. F. McCarthy and T. D. Anagnost. 1998. MUSICFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts. In *ACM Conference on Computer Supported Cooperative Work (CSCW'98)*. 363–372. (Cited on pages 2, 68, 89, and 117.)

K. McCarthy, J. Reilly, L. McGinty, and B. Smyth. 2004a. On the Dynamic Generation of Compound Critiques in Conversational Recommender Systems. In *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 176–184. (Cited on page 47.)

K. McCarthy, J. Reilly, L. Mcginty, and B. Smyth. 2004b. Thinking Positively - Explanatory Feedback for Conversational Recommender Systems. *European Conference on Case-Based Reasoning (ECCBR-04) Explanation Workshop*, 1–10. (Cited on page 35.)

K. McCarthy, M. Salamó, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. 2006. CATS: A Synchronous Approach to Collaborative Group Recommendation. In *FLAIRS Conference*. AAAI Press, 86–91. (Cited on pages 4, 68, 99, and 107.)

S. M. McNee, S. K. Lam, J. A. Konstan, and J. Riedl. 2003. Interfaces for Eliciting New User Preferences in Recommender Systems. In *Proceedings of the 9th International Conference on User Modeling (UM'03)*. Springer-Verlag, Berlin, Heidelberg, 178–187. (Cited on pages 4, 6, and 68.)

S. Mika. 2011. Challenges for Nutrition Recommender Systems *(Workshop Proceedings on Context Aware Intelligent Assistance)*. CEUR-WS.org, 25–33. (Cited on pages 90, 103, and 104.)

B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl. 2004. MovieLens Unplugged: Experiences with a Recommender System on Four Mobile Devices. In *People and Computers XVII — Designing for Society*. Springer London, London, 263–279. (Cited on page 40.)

R. J. Mooney and L. Roy. 2000. Content-based Book Recommending Using Learning for Text Categorization. In *Proceedings of DL-00, 5th ACM Conference on Digital Libraries*. ACM Press, New York, US, San Antonio, US, 195–204. (Cited on page 88.)

K. I. Muhammad, A. Lawlor, and B. Smyth. 2016. A Live-User Study of Opinionated Explanations for Recommender Systems. In *Proceedings of the 21st International Conference on Intelligent User Interfaces (IUI '16)*. ACM, New York, NY, USA, 256–260. (Cited on page 41.)

R. Müller, K. Spang, and S. Ozcan. 2009. Cultural Differences in Decision Making in Project Teams. *International Journal of Managing Projects in Business* 2, 1 (2009), 70–93. (Cited on page 121.)

J. Murphy, C. Hofacker, and R. Mizerski. 2017. Primacy and Recency Effects on Clicking Behavior. *Journal of Computer-Mediated Communication* 11, 2 (07 2017), 522–535. (Cited on page 4.)

S. Najafian and N. Tintarev. 2018. Generating Consensus Explanations for Group Recommendations: An Exploratory Study. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP '18)*. ACM, New York, NY, USA, 245–250. (Cited on pages 6 and 51.)

T. N. Nguyen and F. Ricci. 2017. A Chat-Based Group Recommender System for Tourism. *Information and Communication Technologies.* (2017), 17–30. (Cited on pages 103 and 106.)

G. Ninaus, A. Felfernig, M. Stettinger, S. Reiterer, G. Leitner, L. Weninger, and W. Schanil. 2014. IntelliReq: Intelligent Techniques for Software Requirements Engineering. In *Prestigious Applications of Intelligent Systems (PAIS)*. 1161–1166. (Cited on pages 5, 23, and 44.)

E. Ntoutsi, K. Stefanidis, K. Norvag, and H. P. Kriegel. 2012. Fast Group Recommendations by Applying User Clustering. In *Conceptual Modeling*. Springer Berlin Heidelberg, Berlin, Heidelberg, 126–140. (Cited on pages 36, 37, 48, and 51.)

I. Nunes and D. Jannach. 2017. A Systematic Review and Taxonomy of Explanations in Decision Support and Recommender Systems. *User Modeling and User-Adapted Interaction* 27, 3-5 (Dec. 2017), 393–444. (Cited on page 34.)

M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. 2001. *PolyLens: A Recommender System for Groups of Users*. Springer Netherlands, Dordrecht, 199–218. (Cited on pages 37, 86, 89, 99, and 105.)

J. Pallant. 2007. *SPSS Survival Manual: A Step by Step Guide to Data Analysis Using SPSS for Windows Version 15* (3rd ed.). Open University Press, Milton Keynes, UK, USA. (Cited on pages 58, 61, 62, and 79.)

I. Palomares, L. Martínez, and F. Herrera. 2014a. A Consensus Model to Detect and Manage Non-cooperative Behaviors in Large-Scale Group Decision Making. *IEEE Transactions on Fuzzy Systems* 22, 3 (June 2014), 516 – 530. (Cited on pages 4 and 68.)

I. Palomares, L. Martínez, and F. Herrera. 2014b. MENTOR: A Graphical Monitoring Tool of Preferences Evolution in Large-scale Group Decision Making. *Know.-Based Syst.* 58 (March 2014), 66–74. (Cited on page 38.)

M. J. Pazzani, J. Muramatsu, and D. Billsus. 1996. Syskill and Webert: Identifying Interesting Web Sites. In *AAAI/IAAI, Vol. 1*. AAAI Press / The MIT Press, 54–61. (Cited on page 88.)

J. A. Pereira, P. Matuszyk, S. Krieter, M. Spiliopoulou, and G. Saake. 2018. Personalized recommender systems for product-line configuration processes. *Computer Languages, Systems Structures* (2018). (Cited on page 3.)

W. E. Piper, M. Marrache, R. Lacroix, A. M. Richardsen, and B. D. Jones. 1983. Cohesion as a Basic Bond in Groups. *Human Relations* 36, 2 (1983), 93–108. (Cited on page 119.)

P. Pu and L. Chen. 2007. Trust-inspiring Explanation Interfaces for Recommender Systems. *Knowledge-based Systems* 20, 6 (Aug. 2007), 542–556. (Cited on page 47.)

L. Quijano-Sanchez, J. A. Recio-Garcia, B. Diaz-Agudo, and G. Jimenez-Diaz. 2013. Social Factors in Group Recommender Systems. *ACM Trans. Intell. Syst. Technol.* 4, 1, Article 8 (Feb. 2013), 30 pages. (Cited on page 44.)

L. Quijano-Sanchez, C. Sauer, J. A. Recio-Garcia, and B. Diaz-Agudo. 2017. Make It Personal: A Social Explanation System Applied to Group Recommendations. *Expert Systems with Applications* 76 (Feb. 2017), 36–48. (Cited on pages 34, 36, 48, and 122.)

B. Rastegarpanah, M. Crovella, and K. Gummadi. 2017. Position Paper: Exploring Explanations for Matrix Factorization Recommender Systems. In *FATREC Workshop on Responsible Recommendation at RecSys2017*. OpenBU, 1–4. (Cited on page 48.)

R. Reiter. 1987. A Theory of Diagnosis from First Principles. *Artif. Intell.* 32, 1 (April 1987), 57–95. (Cited on pages 3, 17, and 116.)

S. Reiterer, A. Felfernig, M. Jeran, M. Stettinger, M. Wundara, and W. Eixelsberger. 2015. A Wiki-based Environment for Constraint-based Recommender Systems Applied in the E-Government Domain. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015), Dublin, Ireland, June 29 - July 3, 2015.* 1–10. (Cited on page 5.)

H. J. Reitz. 1977. *Behavior in Organizations*. Richard D. Irwin, Inc., Homewood, IL. (Cited on page 107.)

P. Resnick and R. Sami. 2008. The Information Cost of Manipulation-resistance in Recommender Systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys'08)*. ACM, New York, NY, USA, 147–154. (Cited on page 67.)

F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. 2010. *Recommender Systems Handbook* (1st ed.). Springer-Verlag, Berlin, Heidelberg. (Cited on pages 1, 2, 40, and 88.)

A. Robertson. 2004. *Food and Health in Europe: A New Basis for Action*. WHO Regional Office for Europe. (Cited on pages 85 and 90.)

C. L. Rodriguez and A. F. Brodbeck. 2008. Group Decision Support Systems in Multicultural Context: Potential Influence of Culture in Work Groups. *EnANPAD*. (Cited on page 121.)

A. M. Roza and H. M. Shizgal. 1984. The harris benedict equation reevaluated: resting energy requirements and the body cell mass. *The American Journal of Clinical nutrition* 40, 1 (July 1984), 168–182. (Cited on page 99.)

M. Salamó, K. Mccarthy, and B. Smyth. 2012. Generating Recommendations for Consensus Negotiation in Group Personalization Services. *Personal Ubiquitous Comput.* 16, 5 (June 2012), 597–610. (Cited on page 44.)

T. Sandholm. 2016. *Computing in Mechanism Design*. Palgrave Macmillan UK, London, 1–12. (Cited on page 69.)

M. T. Sanz de Acedo Baquedano, M. C. Elawar, and M. L. Sanz de Acedo Lizárrage. 2007. Factors that affect decision making: gender and age differences. *International Journal of Psychology & Psychological Therapy* 7, 3 (2007), 381–391. (Cited on page 120.)

B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW'01)*. ACM, New York, NY, USA, 285–295. (Cited on page 88.)

H. Schäfer, S. Hors-Fraile, R. P. Karumur, A. Calero Valdez, A. Said, H. Torkamaan, T. Ulmer, and C. Trattner. 2017. Towards Health (Aware) Recommender Systems. In *Proceedings of the 2017 International Conference on Digital Health (DH '17)*. ACM, New York, NY, USA, 157–161. (Cited on pages 1, 7, and 117.)

K. Schmid. 2000. Scoping Software Product Lines: An Analysis of an Emerging Technology. In *Proceedings of the First Conference on Software Product Lines : Experience and Research Directions: Experience and Research Directions*. Kluwer Academic Publishers, Norwell, MA, USA, 513–532. (Cited on page 15.)

M. Schubert, A. Felfernig, and M. Mandl. 2010. FastXplain: Conflict Detection for Constraint-Based Recommendation Problems. In *Trends in Applied Intelligent Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 621–630. (Cited on page 4.)

P. Sedgwick and N. Greenwood. 2015. Understanding the Hawthorne effect. *BMJ* 351 (2015). (Cited on pages 69 and 117.)

C. Senot, D. Kostadinov, M. Bouzid, J. Picault, and A. Aghasaryan. 2017. Evaluation of group profiling strategies. In *In IJCAI 2017*. 2728–2733. (Cited on page 108.)

C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier. 2010. Analysis of Strategies for Building Group Profiles. In *User Modeling, Adaptation, and Personalization*. Springer Berlin Heidelberg, Berlin, Heidelberg, 40–51. (Cited on pages 51, 52, and 60.)

D. Serbos, S. Qi, N. Mamoulis, E. Pitoura, and P. Tsaparas. 2017. Fairness in Package-to-Group Recommendations. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 371–379. (Cited on page 44.)

M. E. Shaw. 1976. *Group dynamics: The Psychology of Small Group Behavior*. McGraw-Hill. (Cited on page 119.)

R. B. Smith, R. Hixon, and B. Horan. 1998. Supporting Flexible Roles in a Shared Space. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW '98)*. ACM, New York, NY, USA, 197–206. (Cited on page 89.)

M. Snooks. 2009. *Health Psychology: Biological, Psychological, and Sociocultural Perspectives*. Jones & Bartlett Learning, Chapter 5: Applications of Health Psychology to Eating Behaviors: Improving health through nutritional changes, 113–144. (Cited on pages 103 and 105.)

F. Sormo, J. Cassens, and A. Aamodt. 2005. Explanation in Case-Based ReasoningPerspectives and Goals. *Expert Syst. Appl.* 24, 2 (Oct. 2005), 109–143. (Cited on page 6.)

M. Stettinger. 2014. Choicla: Towards Domain-independent Decision Support for Groups of Users. In *RecSys'14*. ACM, 425–428. (Cited on pages 16, 21, 45, 50, and 101.)

M. Stettinger, A. Felfernig, G. Leitner, and S. Reiterer. 2015. Counteracting Anchoring Effects in Group Decision Making. In *UMAP (Lecture Notes in Computer Science)*, Vol. 9146. Springer, 118–130. (Cited on pages 3, 4, 6, 44, 68, 69, 90, and 107.)

M. Stolze, S. Field, and P. Kleijer. 2000. Combining Configuration and Evaluation Mechanisms to Support the Selection of Modular Insurance Products. In *Proceedings of the 8th European Conference on Information Systems, Trends in Information and Communication Systems for the 21st Century, ECIS 2000, Vienna, Austria, July 3-5, 2000*. 858–865. (Cited on pages 5 and 23.)

M. Stumptner. 1997. An Overview of Knowledge-Based Configuration. *Artificial Intelligence Community* 10, 2 (April 1997), 111–125. (Cited on pages 5, 9, 15, and 23.)

M. Svensson, J. Laaksolahti, K. Höök, and A. Waern. 2000. A Recipe Based On-line Food Store. In *Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI '00)*. ACM, New York, NY, USA, 260–263. (Cited on pages 87, 91, and 92.)

P. Symeonidis, A. Nanopoulos, and Manolopoulos Y. 2008. Providing Justifications in Recommender Systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 38, 6 (Nov 2008), 1262–1272. (Cited on pages 37 and 43.)

R. Tanner. 2018. Equity Theory - Why Employee Perceptions About Fairness Do Matter. Retrieved May, 2018 from `https://managementisajourney.com/equity-theory-why-employee-perceptions-about-fairness-do-matter/` (Cited on page 64.)

J. C. Teze, S. Gottifredi, A. J. García, and G. R. Simari. 2015. Improving Argumentation-based Recommender Systems through Context-adaptable Selection Criteria. *Expert Syst. Appl.* 42, 21 (2015), 8243–8258. (Cited on page 43.)

E. J. Thomas and C. F. Fink. 1963. Effects of group size. *Psychological Bulletin* 60, 4 (1963), 371–384. (Cited on page 119.)

L. Thompson, L. J. Kray, and E. A. Lind. 1998. Cohesion and Respect: An Examination of Group Decision Making in Social and Escalation Dilemmas. *Journal of Experimental Social Psychology* 34, 3 (1998), 289 – 311. (Cited on page 119.)

P.R Thornton, G. Shaw, and A.M. Williams. 1997. Tourist group holiday decision-making and behaviour: the influence of children. *Tourism Management* 18, 5 (1997), 287–297. (Cited on page 120.)

N. Tintarev. 2009. *Explaining Recommendations*. University of Aberdeen. (Cited on pages 34 and 48.)

N. Tintarev and J. Masthoff. 2007. A Survey of Explanations in Recommender Systems.. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop (ICDEW '07)*. IEEE Computer Society, Washington, DC, USA, 801–810. (Cited on pages 3, 105, 116, and 122.)

N. Tintarev and J. Masthoff. 2011. *Designing and Evaluating Explanations for Recommender Systems*. Springer US, Boston, MA, 479–510. (Cited on pages 33, 34, and 49.)

N. Tintarev and J. Masthoff. 2012. Evaluating the Effectiveness of Explanations for Recommender Systems. *User Modeling and User-Adapted Interaction* 22, 4-5 (Oct. 2012), 399–439. (Cited on pages 33, 36, and 49.)

N. Tintarev and J. Masthoff. 2015. Explaining Recommendations: Design and Evaluation. In *Recommender Systems Handbook*. 353–382. (Cited on page 34.)

N. Tintarev, J. O'Donovan, and Felfernig. 2016. Human Interaction with Artificial Advice Givers: no subtitle. *ACM Transactions on Interactive Intelligent Systems* 6, 4 (31 12 2016), 1–10. (Cited on pages 33 and 34.)

R. W. Toseland, L. V. Jones, and Z. D. Gellis. 2004. *Handbook of Social Work with Groups*. The Guilford Press, Chapter Group Dynamics, 13–31. (Cited on page 124.)

T. N. T. Tran, M. Atas, A. Felfernig, V. M. Le, R. Samer, and M. Stettinger. 2019a. Towards Social Choice-based Explanations in Group Recommender Systems. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '19)*. ACM, New York, NY, USA, 13–21. (Cited on pages 10 and 49.)

T. N. T. Tran, M. Atas, A. Felfernig, R. Samer, and M. Stettinger. 2018b. Investigating Serial Position Effects in Sequential Group Decision Making. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP'18)*. ACM, New York, NY, USA, 239–243. (Cited on pages 12 and 107.)

T. N. T. Tran, M. Atas, A. Felfernig, and M. Stettinger. 2018a. An Overview of Recommender Systems in The Healthy Food Domain. *Journal of Intelligent Information Systems* 50, 3 (01 Jun. 2018), 501–526. (Cited on pages 1, 11, and 85.)

T. N. T. Tran, M. Atas, M. Stettinger, and A. Felfernig. 2016. An extension of CHOICLA User Interfaces for Configurable Products. In *Workshop on Recommender Systems and Big Data Analytics (RS-BDA'16) Graz, Austria, 2016*. 1–4. (Cited on pages 2, 5, 23, and 50.)

T. N. T. Tran, A. Felfernig, V. M. Le, M. Atas, M. Stettinger, and R. Samer. 2019b. User Interfaces for Counteracting Decision Manipulation in Group Recommender Systems. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization, UMAP 2019, Larnaca, Cyprus, June 09-12, 2019*. 93–98. (Cited on pages 11 and 67.)

E. P. K. Tsang. 1993. *Foundations of Constraint Satisfaction.* Academic Press. I–XVIII, 1–421 pages. (Cited on page 16.)

T. Ueta, M. Iwakami, and T. Ito. 2011. A Recipe Recommendation System Based on Automatic Nutrition Information Extraction. In *Proceedings of the 5th International Conference on Knowledge Science, Engineering and Management (KSEM'11)*. Springer-Verlag, Berlin, Heidelberg, 79–90. (Cited on pages 7, 87, 95, 102, and 117.)

T. Ulz, M. Schwarz, A. Felfernig, S. Haas, A. Shehadeh, S. Reiterer, and M. Stettinger. 2017. Human Computation for Constraint-based Recommenders. *Journal of Intelligent Information Systems* 49, 1 (01 Aug 2017), 37–57. (Cited on page 36.)

Y. Van Pinxteren, G. Geleijnse, and P. Kamsteeg. 2011. Deriving a Recipe Similarity Measure for Recommending Healthful Meals. In *Proceedings of the 16th International Conference on Intelligent User Interfaces (IUI '11)*. ACM, New York, NY, USA, 105–114. (Cited on pages 85, 103, 104, and 105.)

B. Van Roy and X. Yan. 2009. Manipulation-resistant Collaborative Filtering Systems. In *Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09)*. ACM, New York, NY, USA, 165–172. (Cited on pages 4, 6, and 68.)

K. Verbert, D. Parra, P. Brusilovsky, and E. Duval. 2013. Visualizing Recommendations to Support Exploration, Transparency and Controllability. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces (IUI '13)*. ACM, New York, NY, USA, 351–362. (Cited on page 34.)

J. Vig, S. Sen, and J. Riedl. 2009. Tagsplanations: Explaining Recommendations using Tags. In *IUI*. ACM, 47–56. (Cited on pages 36 and 40.)

C. Villavicencio, S. Schiaffino, J. A. Diaz-Pace, A. Monteserin, Y. Demazeau, and C. Adam. 2016. A MAS Approach for Group Recommendation Based on Negotiation Techniques. In *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection*. Springer International Publishing, Cham, 219–231. (Cited on page 123.)

W. Wang. 2016. *Enhanced Group Recommender System and Visualization.* Ph.D. Dissertation. University of Technology Sydney, Australia. (Cited on page 6.)

M. Wiesner and D. Pfeifer. 2014. Health Recommender Systems: Concepts, Requirements, Technical Basics and Challenges. *International Journal of Environmental Research and Public Health* 11 (March 2014), 2580–2607. (Cited on pages 1, 7, and 117.)

R. Williams and A. Mcleod. 2008. Consensus Decision-Making. (Cited on page 50.)

D. V. Winterfeldt and W. Edwards. 1986. *Decision Analysis and Behavioral Research*. Cambridge University Press. (Cited on page 43.)

B. Xiao and I. Benbasat. 2007. E-commerce Product Recommendation Agents: Use, Characteristics, and Impact. *MIS Q.* 31, 1 (March 2007), 137–209. (Cited on pages 4, 6, and 68.)

Z. Yu, X. Zhou, Y. Hao, and J. Gu. 2006. TV Program Recommendation for Multiple Viewers Based on User Profile Merging. *User Modeling and User-Adapted Interaction* 16, 1 (March 2006), 63–82. (Cited on page 86.)

S. Zaidi, M. Saif, and A. Zaheer. 2010. The Effect of Workgroup Heterogeneity on Decision Making: An Empirical Investigation. *African Journal of Business Management* 4, 10 (2010), 2132–2139. (Cited on page 120.)

B. Zhang and H. S. Zhou. 2017. Brief Announcement: Statement Voting and Liquid Democracy. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC '17)*. ACM, New York, NY, USA, 359–361. (Cited on pages 9 and 24.)

W. Zhou, J. Wen, Y. S. Koh, Q. Xiong, M. Gao, G. Dobbie, and S. Alam. 2015. Shilling Attacks Detection in Recommender Systems Based on Target Item Analysis. *PLoS ONE* 10 (Jul. 2015), 1–26. (Cited on pages 4, 6, and 68.)