Proceedings of the

RecSys 2012

Workshop on

Human Decision Making in Recommender Systems

(Decisions@RecSys'12)

In conjunction with the

6th ACM Conference on Recommender Systems

September 9-13, 2012, Dublin, Ireland

Preface

Interacting with a recommender system means to take different decisions such as selecting a song/movie from a recommendation list, selecting specific feature values (e.g., camera's size, zoom) as criteria, selecting feedback features to be critiqued in a critiquing based recommendation session, or selecting a repair proposal for inconsistent user preferences when interacting with a knowledge-based recommender. In all these scenarios, users have to solve a decision task.

The complexity of decision tasks, limited cognitive resources of users, and the tendency to keep the overall decision effort as low as possible lead to the phenomenon of bounded rationality, i.e., users exploit decision heuristics rather than trying to take an optimal decision. Furthermore, preferences of users will likely change throughout a recommendation session, i.e., preferences are constructed in a specific decision environment and users do not know their preferences beforehand.

Decision making under bounded rationality is a door opener for different types of non-conscious influences on the decision behavior of a user. Theories from decision psychology and cognitive psychology are trying to explain these influences, for example, decoy effects and defaults can trigger significant shifts in item selection probabilities; in group decision scenarios, the visibility of the preferences of other group members can have a significant impact on the final group decision.

The major goal of this workshop was to establish a platform for industry and academia to present and discuss new ideas and research results that are related to the topic of human decision making in recommender systems. The workshop consisted of technical sessions in which results of ongoing research as reported in these proceedings were presented, a keynote talk given by Joseph A. Konstan on "Decision-Making and Recommender Systems: Failures, Successes, and Research Directions" and a wrap up session chaired by Alexander Felfernig.

Marco de Gemmis, Alexander Felfernig, Pasquale Lops,
Francesco Ricci, Giovanni Semeraro and Martijn Willemsen
September 2012

Workshop Committee

Workshop Co-Chairs

Marco de Gemmis, University of Bari Aldo Moro, Italy
Alexander Felfernig, Graz University of Technology, Austria
Pasquale Lops, University of Bari Aldo Moro, Italy
Francesco Ricci, University of Bozen-Bolzano, Italy
Giovanni Semeraro, University of Bari Aldo Moro, Italy
Martijn Willemsen, Eindhoven University of Technology, Netherlands

Organization

Monika Mandl, Graz University of Technology
Gerald Ninaus, Graz University of Technology

Program Committee

Robin Burke, DePaul University, USA
Li Chen, Hong Kong Baptist University, China
Marco De Gemmis, University of Bari Aldo Moro, Italy
Benedict Dellaert, Erasmus University Rotterdam, Netherlands
Alexander Felfernig, Graz University of Technology, Austria
Gerhard Friedrich, University of Klagenfurt, Austria
Sergiu Gordea, Austrian Institute for Technology, Austria
Andreas Holzinger, Medical University Graz, Austria
Dietmar Jannach, University of Dortmund, Germany
Bart Knijnenburg, University of California, USA
Alfred Kobsa, University of California, USA
Gerhard Leitner, University of Klagenfurt, Austria
Pasquale Lops, University of Bari Aldo Moro, Italy
Walid Maalej, Technische Universität München, Germany
Monika Mandl, Graz University of Technology, Austria
Alexandros Nanopoulos, University of Hildesheim, Germany
Francesco Ricci, University of Bolzano, Italy
Olga C. Santos, UNED, Spain
Giovanni Semeraro, University of Bari Aldo Moro, Italy
Erich Teppan, University of Klagenfurt, Austria
Marc Torrens, Strands, Spain
Martijn Willemsen, Eindhoven University of Technology, Netherlands
Markus Zanker, University of Klagenfurt, Austria

Table of Contents

Copyright © 2012 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

Decision-Making in Recommender Systems: The Role of User's Goals and Bounded Resources

Paolo Cremonesi
Politecnico di Milano
p.zza L.da Vinci 32
Milano, Italy
paolo.cremonesi@polimi.it

Antonio Donatacci
Moviri srl
via Schiaffino 11
Milano, Italy
antonio.donatacci@moviri.com

Franca Garzotto
Politecnico di Milano
p.zza L.da Vinci 32
Milano, Italy
franca.garzotto@polimi.it

Roberto Turrin
Moviri srl
via Schiaffino 11
Milano, Italy
roberto.turrin@moviri.com

ABSTRACT
Many factors that influence users' decision making processes in Recommender Systems (RSs) have been investigated by a relatively vast research of empirical and theoretical nature, mostly in the field of e-commerce. In this paper, we discuss some aspects of the user experience with RSs that may affect the decision making process and outcome, and have been marginally addressed by prior research. These include the nature of *users' goals* and the *dynamic characteristics of the resources space* (e.g., *availability* during the search process). We argue that these subjective and objective factors of the user experience with a RS call for a rethinking of the decision making process as it is normally assumed in traditional RSs, and raise a number or research challenges. These concepts are exemplified in the application domain of on-line *services*, specifically, *hotel booking*- a field where we are carrying on a number of activities in cooperation with a large stakeholder (*Venere.com* – a company of *Expedia* Inc.). Still, most of the arguments discussed in the paper can be extended to other domains, and have general implications for RS design and evaluation.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: Multimedia Systems, User Interfaces. H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval.

General Terms
Design, Empirical Study, Experimentation, Human Factors.

Keywords
Recommender System, decision making, Soft Goal, Bounded Resources, design, evaluation, e-tourism, e-booking

1. INTRODUCTION
Recommender Systems (RSs) help users search large amounts of digital contents and identify more effectively the items – products or services - that are likely to be more attractive or useful. As such, RSs can be characterized as tools that help people making decisions, i.e., make a choice across a vast set of alternatives [12]. A vast amount of research has addressed the problem of how RSs influence users' decision making processes and outcomes. A systematic review of the literature about this topic, focused on e-commerce, is reported by Xiao and Benbasat in [18]. These authors pinpoint that when we regard RSs as decision support tools, the design and evaluation of these systems should take into account other aspects beyond the algorithms that influence users' decision-making processes and outcomes. These aspects are related to individuals' subjective factors as well as the design characteristics of the user experience with the RS. While several theoretical arguments and empirical studies exist that support the positive effects of RA use on decision making quality, research in this field is still inconclusive, highlighting the need for further research.

This paper provides some novel contribution to this research area. Most prior work on RSs for decision support focused on e-commerce domains where users buy on-line products or movies [1]. Our work has instead explore decision making processes in the wide application domain of on-line *services*, specifically, *hotel booking.* We are carrying on a number of activities in close cooperation with a key stakeholder in this field, *Venere.com* (**www.venere.com**). This is a company of the Expedia Inc. group which is leader in online hotel reservations market featuring more than 120,000 hotels, Bed and Breakfasts and vacation rentals in 30,000 destinations worldwide. In this domain, we investigate some subjective aspects of the user experience with RSs - the type of *users' goals*, and some objective, i.e., design related, attributes of RSs – the nature of the *resources space* (e.g., the availability of items along the time in general, and specifically during the search process) that may affect the decision making processes supported by RS. Still, most of our considerations can be extended to other domains, and have implications for research and practice in RS design and evaluation in general.

2. USER GOALS AND "BOUNDED" RESOURCES

2.1 Scenarios
Let us consider the following scenarios, in which the user is engaged with an online hotel reservation system.

Scenario 1. You have to come to Milan and work with your business partners from August 6 to August 10, 2012. You want to reserve a room in a hotel in Milan for that week.

Scenario 2. You will spend a holiday in Milan from September 19 to September 25, 2012, and want to reserve a room.

Scenario 3. You have to attend a business meeting in Milan from September 19 to September 20, 2012, and you need to reserve a room in a hotel in Milan on that dates, for one night

Scenario 4. You are planning a holiday in Central Italy in mid September 2012, and will visit Rome for few days. You need a hotel in that period.

Paper presented at the 2012 Decisions@RecSys workshop in conjunction with the 6th ACM conference on Recommender Systems.
Copyright © 2012 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

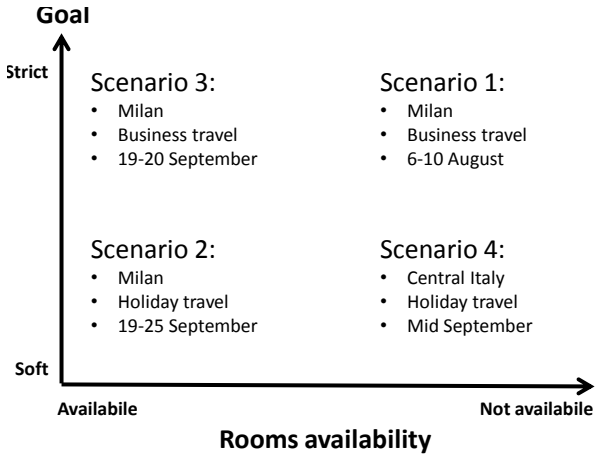


Figure 1: Bounded resources and task goal

How do the above scenarios differ?

In all of them, the user is doing a similar operational task: buying a service, specifically, reserving hotel rooms. Still, there are some significant differences that may influence the decision making processes, and are induced by i) the different nature of the user's goal; ii) the dynamic nature of the services offered by the system the user is interacting with (Figure 1).

In scenarios 1,2 and 3, user's goals are sharp, users' preferences are well defined and have clear-cut criteria for their optimal satisfaction. In scenario 4, the user have less strict preferences – her dates are "flexible", and we may not exclude that she is flexible also with respect to other criteria, or may not know all her preferences beforehand. Preferences are likely to be shaped and changed throughout a session in the specific decision environment. Using the terminology of goal oriented requirements engineering [15], scenario 4 depicts a situation that is characterized by *soft goals* [20], i.e., *open-ended* needs that are progressively elaborated during the interaction with an environment and the decision process, and may be somehow supported by one or more combinations of solutions and decisions[1].

Further differences in the above scenarios are related to the intrinsic nature of resources, in particular, to the *dynamic, time dependent characteristic* of the items in terms of their *availability*.

In scenario 1, the user is making a decision in the context of *a very vast set of stable alternatives*: in the second week of August, hotel availability in Milan is huge, as most people and companies or institutions are on holiday. No matter when and how you reserve a hotel, it is very likely that you will find one that matches you preferences.

In contrast, in scenarios 2, 3, and 4, the user is taking decision in the context of *limited or very limited resources*, or of resources that *become* limited, or even fully *unavailable*, as the decision process proceeds. In scenario 2, the user is looking for hotels in a period - from September 19 to September 25, 2012 – when Milan will host one of the most important international events in the fashion world, the Milan Fashion Week, attracting thousands of people from all over the world. Most hotels are booked one year in advance for that event. Hence, we can reasonably expect that, when searching a room for the whole week, *no hotel is available*.

Scenario 3 considers reservations in the same period of time, but here the user's requirement is less demanding – she is searching a room only for the first day of Milan Fashion week. There might be rooms available on that single date. Still, it may happen that other people are simultaneously trying to make a similar reservation, so that when the user takes her decision, the chosen hotel *is not available any more*.

In scenario 4, the user hasn't decided yet when she exactly will go to Rome, and her dates are flexible. It is likely that she has not specified the reservation period at the beginning of the process, and finds many alternatives matching her preferences on hotel characteristics. Still, the preferred time frame for reservation – mid September – is high season in Rome, and finding a hotel in that period time may be difficult. When she make a specific choice, decides the dates and attempts to make a reservation, the selected hotel may result to be fully booked.

2.2 The decision making process

In all contexts depicted in the above scenarios, the user is facing a problem falling in the class of so called "preferential choice problems" [17], i.e., she needs to take decisions across *an initially vast set of potential alternatives.* In this context, decision making processes are typically modeled as "bounded rationality" phenomena [10]. Bounded rationality – which provides a key theoretical underpinning for RSs – is the notion that, in complex decision-making environments, individuals are often unable to evaluate *all* available alternatives in great depth prior to making their choices, due to the cognitive limitations of their minds, and the finite amount of time they have to make a decision; hence they seek to attain a satisfactory, although not necessarily an optimal, level of achievement, by applying their rationality only after having greatly simplified the set of choices available.

Several authors suggest that the cognitive effort can be reduced with a multiple-stage decision-making process, in which the depth of information processing varies by stage [6][18]. Initially, individuals screen the *complete solution space* (e.g., the set of all hotels featured by the on-line reservation service provider) to identify the (large) set of potential alternatives, or *search set* (e.g., the set of hotels that could be of some interest); then they search through this set, and identify a subset of promising candidates (the *consideration set*). Subsequently, they acquire detailed information on *selected* alternatives to be seriously considered (*in-depth comparison set),* evaluate and compare them in more detail, and finally commit to a specific choice. Although some of the above actions can be iterated, this process is intrinsically linear and it is likely to end with the user making a specific choice and hopefully buying a service.

The same process may not apply exactly in the same terms in the situations described in scenarios 2, 3 and 4 (Figure 2). In scenario 2, the search set is likely to be empty (no hotel is available for the specified period). In scenarios 3 and 4, the search set, the consideration set and the in-depth comparison set are not empty, initially. Still, *their size decreases as the decision process proceed* (e.g., because other users buy some items, or because the user refines her decision criteria, e.g., fixing the dates). Hence, when the user reaches the final step and makes a decision, her choice will likely result unfeasible. In all these cases, after experiencing the unavailability of resources, i.e., of rooms in the desired hotel(s), the user may either *give up* (e.g., she leaves the current on-line reservation service and tries a different one) or *iterate* the

[1] It is worth noticing that soft goals often occur also in entertainment-related domains, such as video-on-demand and interactive TV. For instance, a user may wish to watch a relaxing TV program, without expressing any other requirement

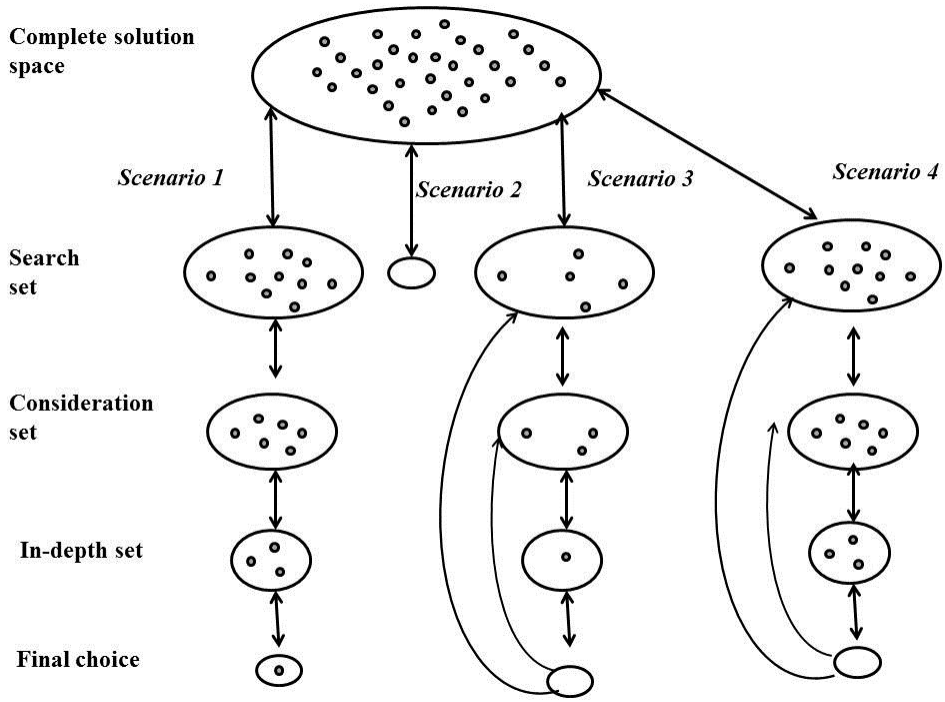


Figure 2: Decision making processes in the four scenarios

process, providing extra input to modify their preferences, exploring the search set, consideration set and in-depth set again, and attempting to make a different decision.

3. CHALLENGES FOR RS DESIGN AND EVALUATION

The examples discussed in the previous section highlight that the decision process in RSs is influenced by the characteristics of both users' goals and the resources meeting users' needs and preferences. How the nature of the goal (sharp or soft) and the dynamic of resources play in the decision making process has been marginally explored in current RS research, and opens a number of research challenges.

A first challenge is to understand the degree at which some key theoretical assumptions underlying most of the existing RSs, such as "bounded rationality", are valid in the context of users' *soft goals*, and how the structure of RS supported decision making processes can be defined in these situations. On the one side, it remains true also that a decision-maker lacks the ability and cognitive resources to arrive at the optimal solution in a vast set of alternatives, and at some point of time she needs to apply her rationality after having greatly simplified the choices available. On the other side, the decision-maker might not be modeled as a "satisfier" - one seeking a satisfactory solution rather than the optimal one, minimizing the cognitive effort - *along the entire decision making process*. At the beginning of the process, the user may indeed be looking for an optimal solution, because her needs and preferences are initially poorly defined, and she does not know yet what the characteristics of such optimal solution are. Hence the initial step of the decision process is more a kind of "sense making" activity that a focused "search": the user is attempting to understand the complexity of the domain and the characteristics of the items in relationship to the specific field of interest, in order to decide what she needs and wants. In this context, the decision making process seems to include a *preliminary phase,* taking place before the progressive elaboration of alternatives, in which the user forges her own preferences, and transforms a soft goal into a sharp goal that characterizes an actual "preferential choice problem". In this preliminary "sense making phase", optimizing cognitive resources and reducing effort might not be an issue, as suggested by some studies [1].

This analysis has challenging implications for RS design in domains where both sharp and soft goals coexist. In these contexts, a designer's goal should go beyond the support to findability – to enable users easily locate what they are precisely looking for – and to the tasks involved in the decision making process as it is conventionally intended. RS design should also support tasks that are essentially *explorative* in nature [1][11], and are oriented towards constructing preferences in the specific domain and decision environment. The challenge is to provide a seamlessly integrated set of interactive design strategies that leverages existing patterns of exploratory interaction, such as faceted navigation and search [13], with existing RS design strategies. It is worth noticing that serendipity can be an important goal also in this exploratory phase and not only when providing recommendations. Promoting crucial contents the existence of which users did not even suspect, so that users can stumble and get interested in them (even if they were not looking for that kind of information), can be as effective (or perhaps more effective) in this phase than in following phases of the decision process.

From a different perspective, also the bounded resources condition challenges existing results concerning the decision making process in typical RSs. The process depicted by [13] and discussed in the previous section applies well in the context of "*unbounded* resources", exemplified by scenario 1 and characterized by a *very vast set of alternatives that remains large when screened and filtered according to user's preference criteria.* In this situation there are theoretical arguments as well as a large number of empirical studies - mostly in the e-commerce

domain [7][8] - that claim that *typical* RSs can provide effective support to users in *all* stages of the decision-making process. They facilitate both the initial screening of available alternatives and the in-depth comparison of item alternatives within the consideration set, reducing the total size of information processed by the users in the search set, consideration set and in-depth search set [13]. Hence we can posit that, *under the unbounded resources condition, typical RSs reduce users' decision effort and users' decision time*, *hence improving the quality of the decision process*. In all cases depicted in scenarios 2, 3 and 4, the decision process is influenced by the "bounded" characteristics of the resources meeting users' needs and preferences, which may affect the validity of the above proposition and the effectiveness of traditional RSs for decision making purposes.

It is well known that, in *any* context, the RS attempt of reducing the user decision effort risks to create the so called *filter bubble* effect. This term, first coined by Eli Pariser in [5] describes a phenomenon in which RSs tend to show only information which agrees with users' past viewpoints, effectively isolating the user in a bubble that tends to exclude items that may be helpful for the users' goals, i.e., novel and serendipitous items. We cannot exclude that potentially negative effects of the bubble phenomenon get amplified in the context of bounded resources: the bubble can result so narrow that, as pinpointed by the discussion in the previous section, the intersection between the bubble and the set of available items is empty. If this is the case, the decision process must be iterated, possibly several times. This situation is likely to *increase users' decision effort and users' decision time*, and *therefore decrease the quality of the decision process.* This in turn have potentially negative effects on the users' perception of on her trust in, usefulness of, and satisfaction with the RS. Even worse, the user may *give up* before completing the decision process, leaving the current on-line reservation service and trying a different one, with obvious implications for the service provider, in terms of customers' trust and actual business outcomes.

In order to overcome these problems, users must be exposed to novel and serendipitous recommendations [2]. This is a paradigmatic shift for the role of RSs in the decision process: from a tool that helps users in narrowing the search set and consideration set in the case of unbounded resources, to a tool that expands the in-depth set in the case of bounded resources.

Defining the *design strategies* of RSs that take into account the possibility of bounded resources is a challenging issue. Some requirements that need to be taking into account are the following:

- Support to decision making processes that are strongly iterative, maximizing the usability of doing and re-doing previous steps, particularly in the re-definition of preferences as the user becomes aware of the lack of available items matching her requirements.
- Need to maintain users' trust [9] and keep the user engaged with the decision process, in spite of the initial failures that potentially can occur because of the lack of resources. In this respect, specific explanation strategies [19] and appropriate conversational interfaces [16] should be defined, which not only improve transparency and explain how recommendations are generated, but also make the user aware of the shortage of resources
- Ability to act both as *filter* that limits the set of valuable alternatives and as *multiplier* that helps the user expand her horizons by recommending serendipitous alternatives.

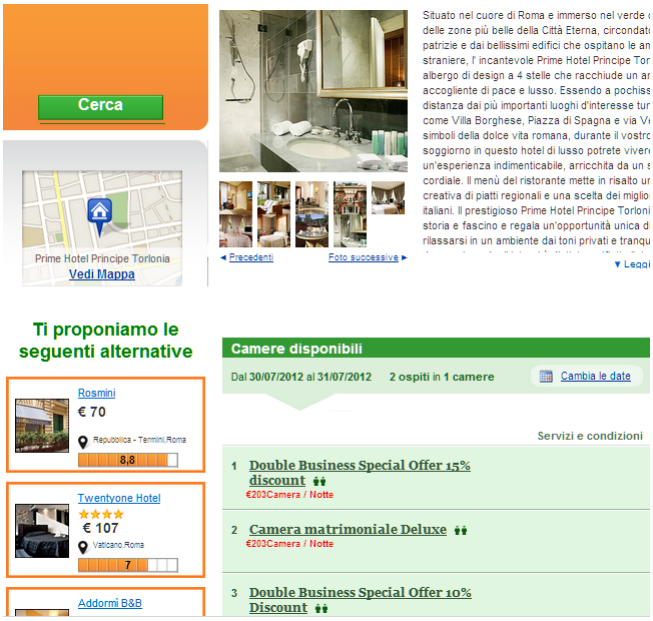


Figure 3: The PoliVenus web application. Recommendations of hotels are on the lower left.

Finally, the concepts of user's goals (sharp or soft) and bounded resources both have implications on *evaluation models, methodologies* and *empirical studies* regarding RSs as decision support tools.

Existing conceptual models for evaluation (e.g., [8][18]) do not provide explicit constructs for users' goals. Previous studies on decision making [5] pinpoint how the nature of users' *tasks* is an important factor affecting individual's behavior and performance. Still, a task as defined in previous studies - "the set of functions that a working person, unit, organization is expected to fulfill or accomplish" [5] - has mainly a functional flavor. Our study emphasizes the need for extending this functional perspective and raising the level of abstraction of the task concept, to address "goals", i.e., broader users' needs. In addition, the discussion presented in the previous sections suggests extensions of existing frameworks for RS evaluation with explicit constructs that address the temporal and dynamic characteristics of RS resources. All these extensions can lead to more powerful *conceptual models* that can help contextualize a wider spectrum of empirical studies in a wide range of RS application domains and situations of use.

4. OUR WORK

Most prior work on RSs for decision support has focused on e-commerce domains where users buy on-line products, pinpointing the influence that different aspects of the user experience with the RS induce on the decision process and outcomes. Our work is currently exploring this issue in a different field, the wide application domain of on-line *services*, such as *hotel booking*. We are working in close cooperation with Venere.com, a company of Expedia Inc. and a key stakeholder in this domain. Our work contains methodological, technical, and empirical innovations.

4.1 Methodology

We have defined a *conceptual model* that provides a more comprehensive framework than the existing ones, and takes into account a number of *new* aspects of the user experience with RSs which have been neglected by previous studies and may significantly influence users' decision-making processes and

#	Question	Possible answers	Area
1	Did you already stayed in the city where the hotel is located?	yes / no	product expertise
2	Did you already stayed in the selected hotel?	yes / no	product expertise
3	Would you have preferred to book a different hotel?	yes / no	decision quality
4	If yes to the previous question, would you have preferred a hotel (more answers are feasible):	cheaper / with more stars / in another city zone / in other dates	decision quality
5	Are you satisfied with your final choice?	not much / fairly / very much	decision quality
6	How much the proposed hotels match your personality?	not at all / fairly / very much	decision quality
7	How long have spent for booking the hotel (minutes)?	5 / 10 / 15 / 20 / 30 / 60	decision effort
8	The time required to choose the hotel is:	reasonable / overmuch / short	decision effort
9	The hotel selection process has been:	easy / hard / very hard	decision effort
10	The range of hotel presented is:	poor / broad / very broad	recommendation quality
11	The set of proposed hotels is:	predictable / with original and unexpected items / very surprising	recommendation quality
12	How much do you think that the characteristics of the reserved hotel will correspond to the real one?	not much / fairly / very much	perceived product risk and trust
13	Do you use online booking systems?	never / sometimes / regularly	profiling
14	If you have used online booking, have you ever used Venere.com to make reservations in the past?	never / sometimes / regularly	profiling
15	Average number of journeys with accommodation per year for holiday purpose		profiling
16	Age, Gender, Nationality, Educational qualification, Occupation		profiling
17	When you travel for holiday, which are the priority criteria with which you choose a hotel?	price / offered services / location / suited for people traveling with me	profiling
18	Where are you in this moment?	home / work / vacation / traveling	context

Figure 4. Questionnaire

outcomes. These include the characteristics of the *goals* – sharp vs. soft – performed with the system (e.g., booking a hotel for vacation or for a business trip) and the *dynamic* characteristics of items (e.g., *availability* during the search process).

4.2 Technical work

We have developed a web-based software framework, *PoliVenus,* for evaluation that facilitates the execution of controlled user studies in this field driven by the constructs of our conceptual model (Figure 3). The framework is based on a *modular* architecture that can be easily customized to different datasets and types of recommender algorithms, and enables researchers to manipulate and control different variables in order to systematically assess the effects of RS use on users' decision making processes.

PoliVenus duplicates all the functionality of the Venere.com on-line booking system, with the exception of payment functions, and contains a catalogue of 6000 accommodations and 500000 users' reviews on the same accommodations. PoliVenus can simulate high-season periods by "reducing" the number of rooms available in a range of selected dates.

Selected users on PoliVenus can be provided with recommendations. Recommendations, in turn, can be provided with any type of algorithm (collaborative and content) from a library of 20 algorithms. Hybrid recommendations can be provided as well, combining any two algorithms. The algorithms have been developed in cooperation with ContentWise[2] (algorithms and datasets can be obtaining by mailing the authors). The user profile is implicitly created by monitoring user's interaction with the "objects" (e.g., pages) describing accommodations.

[2] **www.contentwise.tv**

Recommendations can be provided in different phases of the interaction process (e.g., as alternatives when watching the description page of an accommodation, as a sorting option in a list of hotels).

4.3 Empirical Work

We have designed an experimental setup that allows three different experimental conditions: (a) RS use conditions, (b) bounded resources conditions, (c) RS characteristics, and (d) consumer decision processes.

The first condition is obtained by asking the user to execute one between different tasks, each one representing a different system scenario.

The second condition refers to the configuration of the system, i.e., the possibility to use the application without or with RS support. It should be noticed that this second condition is different from most cases of study discussed by Xiao and Benbasat in [18]. In our implementation, the RS integration doesn't exclude the normal functionalities of the application without RS. This coexistence leads us to reconsider the concept of RS use in our research.

The third condition refers to the possibility to choose a different recommender algorithm among a wide range of recommender algorithms either collaborative, content or hybrid.

The fourth condition allows analyzing the user behavior under limited or unlimited items availability. In our experimental setup, item availability can evolve with time (e.g., the longer is the user decision process, the higher is the probability for the selected item to be unavailable, or the higher is the final price for the selected item).

Therefore we have used the testing environment PoliVenus in a number of preliminary empirical studies, for three key aspects of the bounded resources concept:

- Unavailability: resources may be unavailable for the user (e.g., after selecting hotel and accommodation period, the system informs the user that there are no rooms available).
- Time scarcity: resources may become unavailable as the time passes (e.g., as the user session goes on, the availability of rooms in a hotel decreases).
- Price alteration: prices may change depending on availability of resources (e.g., the system simulates price increase in relation to rooms' availability).

4.4 Participants and Context of Execution

In this section we present the results of a preliminary study executing by using the PoliVenus system. The study was designed as a between subjects controlled experiment, in which we measured the first following experimental conditions, each condition tested with two *independent variables*:

RS use. We have tested two independent variables: (i) with and (ii) without recommendations.

Resources availability. We have tested two independent variables: (i) rooms are always available in any date for any hotel, and (ii) no rooms are available in the first hotel in which the user tries to book, regardless of the dates. We will refer to the two scenarios as *rooms available* and *shortage of rooms*, respectively

We have a total combination of four research variables. We have recruited 15 subjects for each group. Overall, the study involved 60 male users aged between 24 and 50. None of them had been previously used Venere.com.

Each participant was invited to browse the hotel catalog of PoliVenus to search for a double hotel room in Rome and to complete the simulated payment procedure booking the room for two nights. The user was then invited to reply to a set of 18 questions related to the quality of the interaction procedure and satisfaction of the chosen hotel (Figure 4).

4.5 Results

Table 1 presents some results of our preliminary study. Only statistically significant results are presented.

Personalization. The first row of the table summarizes the answers to Q6 in the questionnaire and measure the degree of perceived personalization in the hotels presented to the users during their interaction with the system. As expected, all the users that did not receive recommendations perceived the presented hotels are "not personalized". However, only 10% of the users that did receive recommendations perceived these recommendations as matching their personality.

Table 1: Results

	Without RS	With RS
The proposed hotels match your personality (very much)	0%	10%
Task execution time	5'45''	6'30''
rooms always available	6'00''	6'30''
shortage of rooms	5'30''	6'30''
Consideration set (# of explored hotels)	3.5	11
rooms always available	3	9
shortage of rooms	4	13
Task perceived time	8'40''	8'20''
rooms always available	8'15''	7'40''
shortage of rooms	9'00''	9'00''

Task execution time. The second row of the table estimates users' effort by measuring the time required for the completion of the task. Surprisingly, users receiving recommendations required significantly more time (almost one minute more) then users without recommendations. This results may lead to think that recommendations increase the effort of the decision making process. The last two rows in the table provide a different explanation.

Consideration set. In order to analyze why users receiving recommendations takes longer to complete their task, we have measured how many hotels they explore during their interaction with the system (the consideration set). The third row of the table shows that users receiving recommendations explore a much larger consideration set (almost three times the number of hotels with respect to users not receiving recommendations). This result suggests that recommendations help user to explore a larger number of alternatives. This effect is more evident if we compare the two scenarios "rooms available" and "shortage of rooms". Users not receiving recommendations explore the same small number of hotels, regardless of the difficulties in finding rooms. On the contrary, users receiving recommendations explore twice the number of hotels if there are few rooms available. This suggests that recommendations help users in exploring a larger number of alternatives especially in the scenario of bounded resources.

Perceived time. The last row of the table presents the perceived effort of the decision making process measured with the perceived time for completion of the task (Q7 in the questionnaire). Even if users with recommendations required a significantly longer time to complete their task and explored a much larger number of hotels during their session, their perceived time is the same as the time perceived by user without recommendations. In both cases (with and without recommendations) users dealing with shortage of rooms perceived a longer time for their task, even if the task completion time does not change significantly between the "rooms available" and "shortage of rooms" groups.

5. DISCUSSION AND CONCLUSIONS

The analysis of the results presented in the previous section suggests a number of interesting considerations.

- RSs do not reduce the time required to complete a decision making process. On the contrary, RSs stimulate users to explore more alternatives before making their final choice.
- The effort of the decision making process does not change with the adoption of RSs. Users' perception of the elapsed time is not related to the larger number of explored choices.
- The effort of the decision making process increases in the case of bounded resources. RSs seem not able to alleviate this perceived effort.

Our research has its weaknesses, most notably the limited sample size (60 participants) used for this preliminary test. In spite of the above limitation, our work provides contributions both from a research and practical perspective. To our knowledge, this is the first work that systematically analyzes RSs as decision support systems in the scenario of on-line booking services, focusing of the correlation between resources availability and effectiveness of the recommendations. For the practice of decision support systems design and evaluation, our work may promote further approaches that move beyond the attention to conventional perceived relevance metrics and shift the emphasis to more effort-centric factors.

6. REFERENCES

[1] Bambini, R., Cremonesi, P., Turrin, R., A Recommender System for an IPTV Service Provider: a Real Large-Scale Production Environment. In *Recommender Systems Handbook* , F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, 299-331

[2] Cremonesi P., Garzotto F., Turrin R., Investigating the Persuasion Potential of Recommender Systems from a Quality Perspective: an Empirical Study, ACM Trans. on Interactive Intelligent Systems 2 (2) 2012.

[3] Eierman M.A., F. Niederman, and C. Adams. DSS theory: A model of constructs and relationships. In *Decision Support Systems*, volume 14, 1995, 1–26.

[4] Marchionini, G. Exploratory search: from finding to understanding. CACM, 49 (4), Aprile 2006, 41 – 46.

[5] Pariser. E., *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Press, 2011.

[6] Payne, J. W., Bettman, J. R., and Johnson, E. Adaptive Strategy Selection in Decision Making. J. of Experimental Psychology: Learning, Memory, and Cognition 14 (3), 1988, 534-552

[7] Pereira, R. E. Optimizing Human-Computer Interaction for the Electronic Commerce Environment. J. of Electronic Commerce Research (1:1), 2000, 23-44.

[8] Pu, P., Chen L, and Hu R. A User-Centric Evaluation Framework for Recommender Systems. Proc. RecSys 2011, 157-164

[9] Pu, P. and Chen, L. Trust building with explanation interfaces. In Proc. Intelligent User Interfaces - IUI '06. ACM, 2006, 93–100.

[10] Simon, H.A., Models of bounded rationality. 3. Empirically grounded economic reason, 1997, MIT Press

[11] Spagnolo, L., Bolchini, D., Paolini, P., & Di Blas, N. Beyond Findability. J. of Information Architecture. 2 (1) 2010. 19-36

[12] Swaminathan V.. The impact of recommendation agents on consumer evaluation and choice: The moderating role of category risk, product complexity, and consumer knowledge. J. of Consumer Psychology, 13 (1-2), Feb 2003, 93-101

[13] Todd, P., and Benbasat, I. The Influence of Decision Aids on Choice Strategies: An Experimental Analysis of the Role of Cognitive Effort. Organizational Behavior and Human Decision Processes. 60 (1), 1994, 36-74.

[14] Tunkelang, D.. Faceted Search. In Marchionini, G. (ed.), Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2009

[15] Van Lamsweerde A.. Goal-Oriented Requirements Engineering: A Guided Tour. *Proc. Fifth IEEE Int. Symposium on Requirements Engineering* (RE '01). IEEE, 2001. 249-262

[16] Vappiani P. Pu P. Faltings B, Conversational recommenders with adaptive suggestions. Proc. RecSys 2007, 89-96

[17] Zachary W. A cognitively based functional taxonomy of decision support techniques. Human-Computer Interaction 2 (1), March 1986 25-63

[18] Xiao B. Benbasat I. E-commerce product recommendation agents: use, characteristics, and impact. *Management Information Systems Quarterly* 31 (1) 2007. 137–209

[19] Yoo, K. H. and Gretzel,U. Creating more credible and persuasive recommender systems: The influence of source characteristics on recommender system evaluations. In Ricci F. et al *(eds.) Recommender Systems Handbook.* Springer 2011, 455–477.

[20] Yu, E., Modeling Organizations for Information Systems Requirements Engineering. Proc. 1st International Symposium on Requirements Engineering, RE'93, San Jose, USA, 1993

Enhancement of the Neutrality in Recommendation

Toshihiro Kamishima, Shotaro Akaho, and Hideki Asoh
National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2, Umezono 1-1-1, Tsukuba, Ibaraki, 305-8568 Japan
mail@kamishima.net, s.akaho@aist.go.jp, h.asoh@aist.go.jp

Jun Sakuma
University of Tsukuba
1-1-1 Tennodai, Tsukuba, 305-8577 Japan; and
Japan Science and Technology Agency
4-1-8, Honcho, Kawaguchi, Saitama, 332-0012 Japan
jun@cs.tsukuba.ac.jp

ABSTRACT

This paper proposes an algorithm for making recommendation so that the neutrality toward the viewpoint specified by a user is enhanced. This algorithm is useful for avoiding to make decisions based on biased information. Such a problem is pointed out as the filter bubble, which is the influence in social decisions biased by a personalization technology. To provide such a recommendation, we assume that a user specifies a viewpoint toward which the user want to enforce the neutrality, because recommendation that is neutral from any information is no longer recommendation. Given such a target viewpoint, we implemented information neutral recommendation algorithm by introducing a penalty term to enforce the statistical independence between the target viewpoint and a preference score. We empirically show that our algorithm enhances the independence toward the specified viewpoint by and then demonstrate how sets of recommended items are changed.

Categories and Subject Descriptors

H.3.3 [**INFORMATION SEARCH AND RETRIEVAL**]: Information filtering

Keywords

neutrality, fairness, filter bubble, collaborative filtering, matrix decomposition, information theory

1. INTRODUCTION

A recommender system searches for items and information that would be useful to a user based on the user's behaviors or the features of candidate items [21, 2]. GroupLens [19] and many other recommender systems emerged in the mid-1990s, and further experimental and practical systems have been developed during the explosion of Internet merchandizing. In the past decade, such recommender systems have been introduced and managed at many e-commerce sites to promote items sold at these sites.

Paper presented at the 2012 Decisions@RecSys workshop in conjunction with the 6th ACM conference on Recommender Systems. Copyright © 2012 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.
2012 Decisions@RecSys workshop September 9, 2012, Dublin, Ireland

The influence of personalization technologies such as recommender systems or personalized search engines on people's decision making is getting stronger and stronger. For example, at a shopping site, if a customer checks a recommendation list and finds a five-star-rated item, he/she would more seriously consider buying the highly rated item. The filter bubble, which is the selection of the appropriate diversity of information provided to users, is one of problems accompanying the growing influence of recommendation or personalization.

The problem of the filter bubble was recently posed by Pariser [17]. Via the influence of personalized technologies, the topics of information provided to users are becoming restricted to those originally preferred by them, and this restriction is not notified by users. This situation is compared to shutting up each individual in a separate *bubble*. Pariser claimed that due to the obstruction of these bubbles, users lose the opportunity of finding new topics and that sharing public matters throughout our society is getting harder. To discuss this filter bubble problem, a panel discussion was held at the RecSys 2011 conference [20].

During the RecSys panel discussion, panelists made the following assertions about the filter bubble problem. Biased topics would be certainly selected by the influence of personalization, but at the same time, it would be intrinsically impossible to make recommendations that are absolutely neutral from any viewpoint, and the diversity of provided topics intrinsically has a trade-off relation to the fitness of these topics for users' interests or needs. To recommend something, or more generally to select something, one must consider the specific aspect of a thing and must ignore the other aspects of the thing. The panelists also pointed out that current recommender systems fail to satisfy users' information need that they search for a wide variety of topics in the long term.

To solve this problem, we propose an information neutral recommender system that guarantees the neutrality of recommendation results. As pointed out during the RecSys 2011 panel discussion, because it is impossible to make a recommendation that is absolutely neutral from all viewpoints, we consider neutrality from the viewpoint or information specified by a user. For example, users can specify a feature of an item, such as a brand, or a user feature, such as a gender or an age, as a viewpoint. An information neu-

tral recommender system is designed so that these specified features will not affect recommendation results. This system can also be used to avoid the use of information that is restricted by law or regulation. For example, the use of some information is prohibited for the purpose of making recommendation by privacy policies.

We borrowed the idea of fairness-aware mining, which we proposed earlier [11], to build this information neutral recommender system. To enhance the neutrality or the independence in recommendation, we introduce a regularization term that represents the mutual information between a recommendation result and the specified viewpoint.

Our contributions are as follows. First, we present a definition of neutrality in recommendation based on the consideration of why it is impossible to achieve an absolutely neutral recommendation. Second, we propose a method to enhance the neutrality that we defined and combine it with a latent factor recommendation model. Finally, we demonstrate that the neutrality of recommendation can be enhanced and how recommendation results change by enhancing the neutrality.

In section 2, we discuss the filter bubble problem and neutrality in recommendation and define the goal of an information neutral recommender task. An information neutral recommender system is proposed in section 3, and its experimental results are shown in section 4. Sections 5 and 6 cover related work and our conclusion, respectively.

2. INFORMATION NEUTRALITY

In this section, we discuss information neutrality in recommendation based on the considerations on the filter bubble problem and the ugly duckling theorem.

2.1 The Filter Bubble Problem

We here summarize the filter bubble problem posed by Pariser and the discussion in the panel about this problem at the RecSys 2011 conference. The *filter bubble* problem is a concern that personalization technologies, including recommender systems, narrow and bias the topics of information provided to people while they don't notice these facts [17].

Pariser demonstrated the following examples in a TED talk about this problem [16]. In a social network service, Facebook[1], users have to specify a group of *friends* with whom they can chat or have private discussions. To help users find their friends, the service has a function to list other users' accounts that are expected to be related to a user. When Pariser started to use Facebook, the system showed a recommendation list that consisted of both conservative and progressive people. However, because he has more frequently selected progressive people as friends, conservative people have been excluded from his recommendation list by a personalization functionality. Pariser claimed that the system excluded conservative people without his permission and that he lost the opportunity of getting a wide variety of opinions.

He furthermore demonstrated a collection of search results from Google[2] for the query "Egypt" during the Egyptian uprising in 2011 from various people. Even though such a highly important event was occurring, only sightseeing pages were listed for some users instead of news pages about the Egyptian uprising, due to the influence of personalization. In this example, he claimed that personalization technology spoiled the opportunity to obtain information that should be commonly shared in our society.

We consider that Pariser's claims can be summarized as follows. The first point is the problem that users lost opportunities to obtain information about a wide variety of topics. A chance to know things that could make users' lives fruitful was lessened. The second point is the problem that each individual obtains information that is too personalized, and thus the amount of shared information is decreased. Pariser claimed that the loss of sharing information is a serious obstacle for building consensus in our society. He claimed that the loss of the ability to share information is a serious obstacle for building consensus in our society.

RecSys 2011, which is a conference on recommender systems, held a panel discussion the topic of which was this filter bubble problem [20]. This panel concentrated on the following three arguing points. (1) Are there *filter bubbles*? Resnick pointed out the possibility that personalization technologies narrow users' experience in the mid 1990s. Because selecting specific information by definition leads to ignoring other information, the diversity of users' experiences intrinsically have a trade-off relation to the fitness of information for users' interests. As seen in the difference between the perspective of al-Jazeera and that of Fox News, this problem exists irrespective of personalization. Further, given signals or expressions of users' interest, it is difficult to adjust how much a system should meet those interests.

(2) To what degree is personalized filtering a problem? There is no absolutely neutral viewpoint. On the other hand, the use of personalized filtering is inevitable, because it is not feasible to exhaustively access the vast amount of information in the universe. One potential concern is the effect of selective exposure, which is the tendency to get reinforcement of what people already believe. According to the results of studies about this concern, this is not so serious, because people viewing extreme sites spend more time on mainstream news as well.

(3) What should we as a community do to address the filter bubble issue? To adjust the trade-off between diversity and fitness of information, a system should consider users' immediate needs as well as their long-term needs. Instead of selecting individual items separately, a recommendation list or portfolio should be optimized as a whole.

2.2 The Neutrality in Recommendation

The absence of the absolutely neutral viewpoint is pointed out in the above panel. We here more formally discuss this point based on the ugly duckling theorem.

The *ugly duckling theorem* is a classical theorem in pattern recognition literature that asserts the impossibility of classi-

[1]http://www.facebook.com

[2]http://www.google.com

fication without weighing certain features or aspects of objects against the others [26]. Consider a case that n ducklings are represented by at least $\log_2 n$ binary features, for example, black feathers or a fat body, and are classified into positive or negative classes based on these features. If the positive class is represented by Boolean functions of binary features, it is easy to prove that the number of possible functions that classify an arbitrary pair of ducklings into a positive class is 2^{n-2}, even if choosing any pairs of ducklings. Provided that the similarity between a pair of ducklings is measured by the number of functions that classify them into the same class, the similarity between an ugly duckling and an arbitrary normal duckling is equal to the similarity between any pair of ducklings. In other words, an ugly duckling looks like a normal duckling.

Why is an ugly duckling ugly? As described above, an ugly duckling is as ugly as a normal duckling, if all features and functions are treated equally. The attention to an arbitrary feature such as black feathers makes an ugly duckling ugly. When we classify something, we of necessity pay attention to certain features, aspects, or viewpoints of classified objects. Because recommendation is considered as a task to classify items into a relevant class or an irrelevant one, certain features or viewpoints must be inevitably weighed when making recommendation. Consequently, the absolutely neutral recommendation is intrinsically impossible.

We propose a neutral recommendation task other than the absolutely neutral recommendation. Recalling the ugly duckling theorem, we must focus on certain features or viewpoints in classification. This fact indicates that it is feasible to make a recommendation that is neutral from a specific viewpoint instead of all viewpoints. We hence advocate an *Information Neutral Recommender System* that enhances the neutrality in recommendation from the viewpoint specified by a user. In the case of Pariser's Facebook example, a system enhances the neutrality so that recommended friends are conservative or progressive, but the system is allowed to make biased decisions in terms of the other viewpoints, for example, the birthplace or age of friends.

3. AN INFORMATION NEUTRAL RECOMMENDER SYSTEM

In this section, we formalize a task of information neutral recommendation and show a solution algorithm for this task.

3.1 Task Formalization

In [8], recommendation tasks are classified into *Recommending Good Items* that meet a user's interest, *Optimizing Utility* of users, and *Predicting Ratings* of items for a user. Among these tasks, we here concentrate on the task of predicting ratings.

We formalize an information neutral variant of a predicting ratings task. $x \in \{1, \ldots, n\}$ and $y \in \{1, \ldots, m\}$ denote a user and an item, respectively. An event (x, y) is a pair of a specific user x and a specific item y. Here, s denotes a rating value of y as given by x. We here assume that the domain of ratings is real values, though domain of ratings is commonly a set of discrete values, e.g., $\{1, \ldots, 5\}$. These variables are common for an original predicting ratings task.

To treat the information neutrality in recommendation, we additionally introduce a viewpoint variable, v, which indicates a viewpoint neutrality from which is enhanced. This variable is specified by a user, and its value depends on an event. Possible examples of a viewpoint variable are a user's gender, which depends on a user part of an event, movie's release year, which depends on an item's part of an event, and a timestamp when a user rates an item, which depends on both elements in an event. In this paper, we restrict the domain of a viewpoint variable to a binary type, $0, 1$, but it is easy to extend to a multinomial case. An example consists of an event, (x, y), a rating value for the event, s, and a viewpoint value for the event, v. A training set is a set of N examples, $\mathcal{D} = \{(x_i, y_i, s_i, v_i)\},\ i = 1, \ldots, N$.

Given a new event, (x, y), and its corresponding viewpoint value, v, a rating prediction function, $\hat{s}(x, y, v)$, predicts a rating value of an item y by a user x. While this rating prediction function is estimated in our task setting, a loss function, $\mathrm{loss}(s^*, \hat{s})$, and a neutrality function, $\mathrm{neutral}(\hat{s}, v)$, are given as task inputs. A loss function represents the dissimilarity between a true rating value, s^*, and a predicted rating value, $\hat{s}$. A neutrality function quantifies the degree of the neutrality of a rating value from a viewpoint expressed by a viewpoint variable. Given a training set, $\mathcal{D}$, a goal of an information neutral recommendation (predicting rating case) is to acquire a rating prediction function, $\hat{s}(x, y, t)$, so that the expected value of a loss function is as small as possible and the expected value of a neutral function is as large as possible over (x, y, v). We formulate this goal by finding a rating prediction function, $\hat{s}$, so as to minimize the following objective function:

$$\mathrm{loss}(s^*, \hat{s}(x, y, v)) - \eta\, \mathrm{neutral}(\hat{s}(x, y, v), v), \tag{1}$$

where $\eta > 0$ is a parameter to balance between the loss and the neutrality.

3.2 A Prediction Model

In this paper, we adopt a latent factor model for predicting ratings. This latent factor model, which is a kind of a matrix decomposition model, is defined as equation (3) in [12], as follows:

$$\hat{s}(x, y) = \mu + b_x + c_y + \mathbf{p}_x \mathbf{q}_y^\top, \tag{2}$$

where μ, b_x, and c_y are global, per user, and per item bias parameters, respectively, and $\mathbf{p}_x$ and $\mathbf{q}_y$ are K-dimensional parameter vectors, which represent the cross effects between users and items. We adopt a squared loss as a loss function. As a result, parameters of a rating prediction function can be estimated by minimizing the following objective function:

$$\sum_{(x_i, y_i, s_i) \in \mathcal{D}} (s_i - \hat{s}(x_i, y_i))^2 + \lambda R, \tag{3}$$

where R represents an L_2 regularizer for parameters b_x, c_y, $\mathbf{p}_x$, and $\mathbf{q}_y$, and λ is a regularization parameter. Once we learned the parameters of a rating prediction function, we can predict a rating value for any event by applying equation (2).

We then extend this model to enhance the information neutrality. First, we modify the model of equation (2) so as to depend on the value of a viewpoint variable, v. For each value of v, 0 and 1, we prepare a parameter set, $\mu^{(v)}$, $b_x^{(v)}$,

$c_y^{(v)}$, $\mathbf{p}_x^{(v)}$, and $\mathbf{q}_y^{(v)}$. One of parameter sets is chosen according as a value of v, and we get a rating prediction function:

$$\hat{s}(x, y, v) = \mu^{(v)} + b_x^{(v)} + c_y^{(v)} + \mathbf{p}_x^{(v)} {\mathbf{q}_y^{(v)}}^\top. \quad (4)$$

We next define a neutrality function to quantify the degree of the information neutrality from a viewpoint variable, v. In this paper, we borrow an idea from [11] and quantify the degree of the information neutrality by negative mutual information under the assumption that neutrality is regarded as statistical independence. A neutrality function is defined as:

$$\begin{aligned} -I(\hat{s}; v) &= \sum_{v \in \{0,1\}} \int \Pr[\hat{s}, v] \log \frac{\Pr[\hat{s}|v]}{\Pr[\hat{s}]} d\hat{s} \\ &= \sum_{v \in \{0,1\}} \Pr[v] \int \Pr[\hat{s}|v] \log \frac{\Pr[\hat{s}|v]}{\Pr[\hat{s}]} d\hat{s}. \end{aligned} \quad (5)$$

The marginalization over v is then replaced with the sample mean over a training set, $\mathcal{D}$, and we get

$$\frac{1}{N} \sum_{(v) \in \mathcal{D}} \int \Pr[\hat{s}|v] \log \frac{\Pr[\hat{s}|v]}{\Pr[\hat{s}]} d\hat{s}. \quad (6)$$

Note that $\Pr[\hat{s}]$ can be calculated by $\sum_v \Pr[\hat{s}|v] \Pr[v]$, and we use a sample mass function as $\Pr[v]$.

Now, all that we have to do is to compute distribution $\Pr[\hat{s}|v]$, but this computation is difficult. This is because a value of a function $\hat{s}$ is not probabilistic but rather deterministic depending on x, y, and v; and thus distribution $\Pr[\hat{s}|x, y, v]$ has a form of collection of Dirac's delta functions, $\delta(\hat{s}(x, y, v))$. $\Pr[\hat{s}|v]$ can be obtained by marginalizing this distribution over x and y. As a result, $\Pr[\hat{s}|v]$ also becomes a hyper function like $\Pr[\hat{s}|x, y, v]$, and it is not easy to manipulate. We therefore introduce a histogram model to represent $\Pr[\hat{s}|v]$. Values of predicted ratings, $\hat{s}$, are divided into bins, because sample ratings are generally discrete. The distribution $\tilde{\Pr}[\hat{s}|v]$ is expressed by a histogram model. By replacing $\Pr[\hat{s}|v]$ with $\tilde{\Pr}[\hat{s}|v]$, equation (6) becomes

$$\frac{1}{N} \sum_{(v) \in \mathcal{D}} \sum_{\hat{s} \in \mathrm{Bin}} \tilde{\Pr}[\hat{s}|v] \log \frac{\tilde{\Pr}[\hat{s}|v]}{\tilde{\Pr}[\hat{s}]}, \quad (7)$$

where Bin denotes a set of bins of a histogram. Note that because a distribution function, $\Pr[\hat{s}|v]$, is replaced with a probability mass function, $\tilde{\Pr}[\hat{s}|v]$, an integration over $\hat{s}$ is replaced with the summation over bins.

By substituting equations (4) and (7) into equation (1) and adopting a squared loss function as in the original latent factor case, we obtain an objective function of an information neutral recommendation model:

$$\mathcal{L}(\mathcal{D}) = \sum_{(x_i, y_i, s_i, v_i) \in \mathcal{D}} (s_i - \hat{s}(x_i, y_i, v_i))^2 + \eta\, I(\hat{s}; v) + \lambda\, R, \quad (8)$$

where a regularization term, R, is a sum of L_2 regularizers of parameter sets for each value of v. Model parameters, $\{\mu^{(v)}, b_x^{(v)}, c_y^{(v)}, \mathbf{p}_x^{(v)}, \mathbf{q}_y^{(v)}\}, v \in \{0, 1\}$, are estimated so as to minimize this objective function. However, it is very difficult to derive an analytical form of gradients of this objective function, because the histogram transformation used for expressing $\Pr[\hat{s}|v]$ is too complicated. We therefore adopt the Powell optimization method, because it can be applied without computing gradients.

4. EXPERIMENTS

We implemented our information neutral recommender system in the previous section and applied it to a benchmark data set.

4.1 A Data Set

We used a Movielens 100k data set [7] in our experiments. As described in section 3.2, we adopted the Powell method for optimizing an objective function. Unfortunately, this method is too slow to apply to a large data set, because the number of evaluation times of an objective function becomes very large to avoid the computation of gradients. Therefore, we shrank the Movielens data set by extracting events whose user ID and item ID were less than or equal to 200 and 300, respectively. This shrunken data set contained 9,409 events, 200 users, and 300 items.

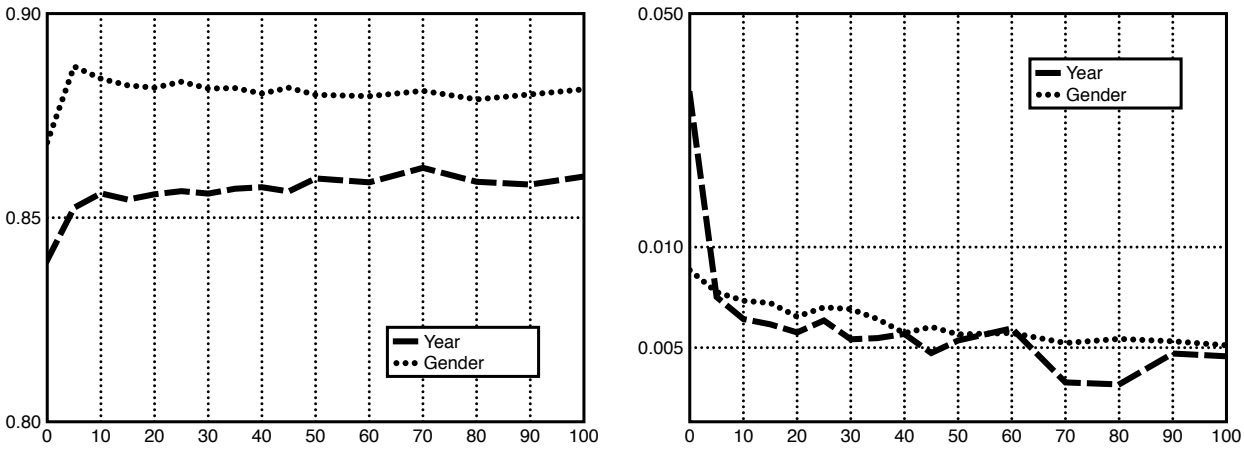
We tested the following two types of viewpoint variable. The first type of variable, Year, represents whether a movie's release year is newer than 1990, which depends on an item part of an event. In [12], Koren reported that the older movies have a tendency to be rated higher, perhaps because only masterpieces have survived. When adopting Year as a viewpoint variable, our recommender enhances the neutrality from this masterpiece bias. The second type of variable, Gender, represents the user's gender, which depends on the user part of an event. The movie rating would depend on the user's gender, and our recommender enhances the neutrality from this factor.

4.2 Experimental Conditions

We used the implementation of the Powell method in the SciPy package [22] as an optimizer for an objective function (8). To initialize parameter, events in a training set, $\mathcal{D}$, were first divided into two sets according to their viewpoint values. For each value of a viewpoint variable, parameters are initialized by minimizing an objective function of an original latent factor model (equation (3)). For the convenience in implementation, a loss term of an objective was scaled by dividing it by the number of training examples, and an L_2 regularizer was scaled by dividing it by the number of parameters. We use a regularization parameter $\lambda = 0.01$ and the number of latent factors, $K = 1$, which are the lengths of vectors $\mathbf{p}^{(v)}$ or $\mathbf{q}^{(v)}$. Because the original rating values are $1, 2, \ldots, 5$, we adopted five bins whose centers are $1, 2, \ldots, 5$, in equation (7). We performed a five-fold cross-validation procedure to obtain evaluation indices of the prediction accuracy and the neutrality from a viewpoint variable.

4.3 Experimental Results

Experimental results are shown in Figure 1. Figure 1(a) shows the changes of prediction errors measured by a mean absolute error (MAE) index. The smaller value of this index indicates better prediction accuracy. Figure 1(b) shows the changes of the mutual information between predicted ratings and viewpoint values. The smaller mutual information indicates a higher level of neutrality. Mutual information is normalized into the range $[0, 1]$ by the method of employing the geometrical mean in [24]. Note that distribution $\Pr[\hat{s}|v]$



(a) mean absolute error (MAE)

(b) normalized mutual information (NMI)

Figure 1: Changes of the degrees of neutrality accompanying the increase of a neutrality parameter

NOTE : Figure 1(a) shows the changes of prediction errors measured by a mean absolute error (MAE) index. The smaller value of this index indicates better prediction accuracy. Figure 1(b) shows the changes of the mutual information between predicted ratings and viewpoint values. The smaller mutual information indicates a higher level of neutrality. The X-axes of these figures represent parameter values of η. Dashed lines and dotted lines show the results using Year and Gender as viewpoint variables, respectively.

is required to compute this mutual information, and we used the same histogram model as in equation (7). The X-axes of these figures represent parameter values of η, which balance the prediction accuracy and the neutrality. These parameters were changed from 0, at which the neutrality term was completely ignored, to 100, at which the neutrality was highly emphasized. Dashed lines and dotted lines show the results using Year and Gender as viewpoint variables, respectively.

MAE was 0.90, when offering a mean score, 3.74, for all users and all items. In Figure 1(a), MAEs were better than this baseline, which is perfectly neutral from all viewpoints. Furthermore, the increase of MAEs as the neutrality parameter, η, was not so serious. Turning to the Figure 1(b), this demonstrates that the neutrality is enhanced as the neutrality parameter, η, increases from both viewpoints, Year and Gender. By drawing attention to the fact that the Y-axis is logarithmic, we can conclude that an information neutrality term is highly effective. In summary, our information neutral recommender system successfully enhanced the neutrality without seriously sacrificing the prediction accuracy.

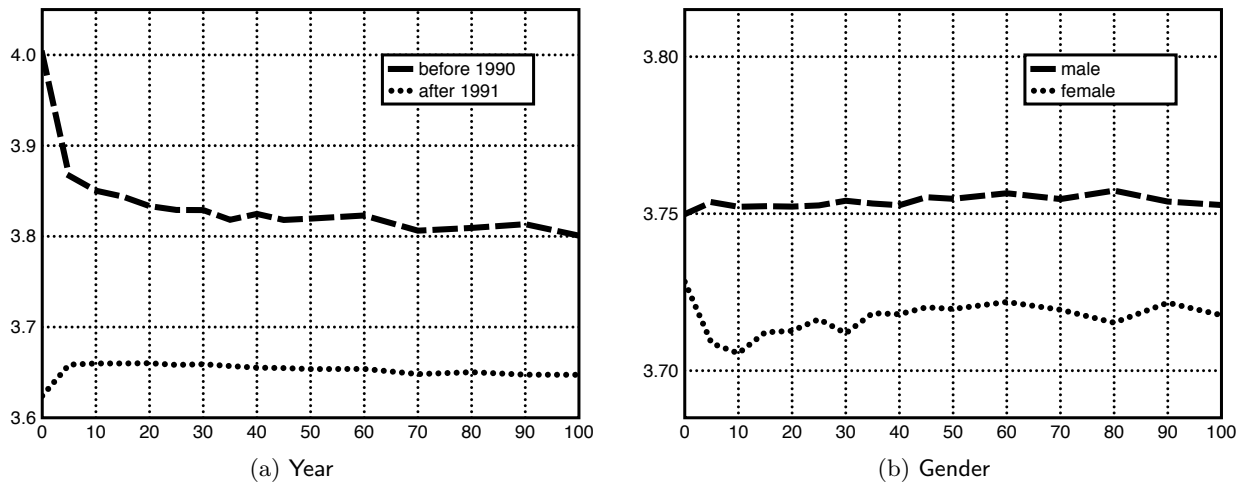
Figure 2 shows the changes of mean predicted scores. In both figures, the X-axes represent parameter values of η, and the Y-axes represent mean predicted scores for each case of using different viewpoint value. Figure 1(a) shows mean predicted scores when a viewpoint variable is Year. Dashed and dotted lines show the results under the condition a viewpoint variable is "before 1990" and "after 1991", respectively. Figure 1(b) shows mean predicted scores when a viewpoint variable is Gender. Dashed and dotted lines show the results obtained by setting a viewpoint to "male" and "female", respectively.

We first discuss a case that a viewpoint variable is Year. According to Figure 1(b), neutrality was drastically improved in the interval that η is between 0 and 10. By observing the corresponding interval in Figure 2, two lines that were obtained for different viewpoints became close each other. This means that prediction scores become less affected by a viewpoint value, and this corresponds the improvement of neutrality. After this range, the decrease of NMI became smaller in Figure 1(b), and the lines in the corresponding interval in Figure 2 were nearly parallel. This indicated that the difference between two score sequences less changes, and the improvement in neutrality did too. We move on to a Gender case. By comparing the changes of NMI between Year and Gender cases in Figure 1(b), the decrease of NMI in a Gender case was much smaller than that of a Year case. This phenomenon could be confirmed by the fact that two lines were nearly parallel in Figure 2(b). This is probably because the score differences in a Gender case are much smaller than those in a Year at the point $\eta = 0$, and there is less margin for improvement. Further investigation will be required in this point.

5. RELATED WORK

To enhance the neutrality, we borrowed an idea from our previous work [11], which is an analysis technique for fairness/discrimination-aware mining. Fairness/discrimination-aware mining is a general term for mining techniques designed so that sensitive information does not influence mining results. In [18], Pedreschi et al. first advocated such mining techniques, which emphasized the unfairness in association rules whose consequents include serious determinations. Like this work, a few techniques for detecting unfair treatments in mining results have been proposed [14, 25]. These techniques might be useful for detecting biases in recommendation.

Another type of fairness-aware mining technique focuses on classification designed so that the influence of sensitive information to classification results is reduced [11, 3, 10] These



(a) Year (b) Gender

Figure 2: Changes of mean predicted scores accompanying the increase of a neutrality parameter

NOTE : In both figures, the X-axes represent parameter values of η, and the Y-axes represent mean predicted scores for each case of different viewpoint value. Figure 1(a) shows mean predicted scores when a viewpoint variable is Year. Dashed and dotted lines show the results under the condition a viewpoint variable is "before 1990" and "after 1991", respectively. Figure 1(b) shows mean predicted scores when a viewpoint variable is Gender. Dashed and dotted lines show the results obtained by setting a viewpoint to "male" and "female", respectively.

techniques would be directly useful in the development of an information neutral variant of content-based recommender systems, because content-based recommenders can be implemented by adopting classifiers.

Information neutrality can be considered as diversity in recommendation in a broad sense. McNee et al. pointed out the importance of factors other than prediction accuracy, including diversity, in recommendation [15]. Topic diversification is a technique for enhancing the diversity in a recommendation list [27]. Smyth et al. proposed a method for changing the diversity in a recommendation list based on a user's feedback [23].

There are several reports about the influence of recommendations on the diversity of items accepted by users. Celma et al. reported that recommender systems have a popularity bias such that popular items have a tendency to be recommended more and more frequently [4]. Fleder et al. investigated the relation between recommender systems and their impact on sales diversity by simulation [6]. Levy et al. reported that sales diversity could be slightly enriched by recommending very unpopular items [13].

Because information neutral recommenders can be used to avoid the exploitation of private information, these techniques are related to privacy-preserving data mining [1]. Independent component analysis might be used to maintain the independence between viewpoint values and recommendation results [9]. In a broad sense, information neutral recommenders are a kind of cost-sensitive learning technique [5], because these recommenders are designed to take into account the costs of enhancing the neutrality.

6. CONCLUSION

In this paper, we proposed an information neutral recommender system that enhanced neutrality from a viewpoint specified by a user. This system is useful for alleviating the filter bubble problem, which is a concern that personalization technologies narrow users' experience. We then developed an information neutral recommendation algorithm by introducing a regularization term that quantifies neutrality by mutual information between a predicted rating and a viewpoint variable expressing a user's viewpoint. We finally demonstrated that neutrality could be enhanced without sacrificing prediction accuracy by our algorithm.

The most serious issue of our current algorithm is scalability. This is mainly due to the difficulty in deriving the analytical form of gradients of an objective function. We plan to develop another objective function whose gradients can be derived analytically. The degree of statistical independence is currently quantified by mutual information. We want to test other indexes, such as kurtosis, which are used for independent component analysis. We will develop an information neutral version of other recommendation models, such as pLSI/LDA or nearest neighbor models.

7. ACKNOWLEDGMENTS

We would like to thank for providing a data set for the Grouplens research lab. This work is supported by MEXT/JSPS KAKENHI Grant Number 16700157, 21500154, 22500142, 23240043, and 24500194, and JST PRESTO 09152492.

8. REFERENCES

[1] C. C. Aggarwal and P. S. Yu, editors. *Privacy-Preserving Data Mining: Models and Algorithms*. Springer, 2008.

[2] J. Ben Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5:115–153, 2001.

[3] T. Calders and S. Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21:277–292, 2010.

[4] Ò. Celma and P. Cano. From hits to niches?: or how popular artists can bias music recommendation and discovery. In *Proc. of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, 2008.

[5] C. Elkan. The foundations of cost-sensitive learning. In *Proc. of the 17th Int'l Joint Conf. on Artificial Intelligence*, pages 973–978, 2001.

[6] D. Fleder and K. Hosanagar. Recommender systems and their impact on sales diversity. In *ACM Conference on Electronic Commerce*, pages 192–199, 2007.

[7] Grouplens research lab, university of minnesota. ⟨`http://www.grouplens.org/`⟩.

[8] A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10:2935–2962, 2009.

[9] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, 2001.

[10] F. Kamiran, T. Calders, and M. Pechenizkiy. Discrimination aware decision tree learning. In *Proc. of the 10th IEEE Int'l Conf. on Data Mining*, pages 869–874, 2010.

[11] T. Kamishima, S. Akaho, and J. Sakuma. Fairness-aware learning through regularization approach. In *Proc. of The 3rd IEEE Int'l Workshop on Privacy Aspects of Data Mining*, pages 643–650, 2011.

[12] Y. Koren. Collaborative filtering with temporal dynamics. In *Proc. of the 15th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 447–455, 2009.

[13] M. Levy and K. Bosteels. Music recommendation and the long tail. In *WOMRAD 2010: Recsys 2010 Workshop on Music Recommendation and Discovery*, 2010.

[14] B. T. Luong, S. Ruggieri, and F. Turini. k-NN as an implementation of situation testing for discrimination discovery and prevention. In *Proc. of the 17th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 502–510, 2011.

[15] S. M. McNee, J. Riedl, and J. A. Konstan. Accurate is not always good: How accuracy metrics have hurt recommender systems. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 1097–1101, 2006.

[16] E. Pariser. The filter bubble. ⟨`http://www.thefilterbubble.com/`⟩.

[17] E. Pariser. *The Filter Bubble: What The Internet Is Hiding From You*. Viking, 2011.

[18] D. Pedreschi, S. Ruggieri, and F. Turini. Discrimination-aware data mining. In *Proc. of the 14th Int'l Conf. on Knowledge Discovery and Data Mining*, 2008.

[19] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of Netnews. In *Proc. of the Conf. on Computer Supported Cooperative Work*, pages 175–186, 1994.

[20] P. Resnick, J. Konstan, and A. Jameson. Panel on the filter bubble. The 5th ACM conference on Recommender systems, 2011. ⟨`http://acmrecsys.wordpress.com/2011/10/25/panel-on-the-filter-bubble/`⟩.

[21] P. Resnick and H. R. Varian. Recommender systems. *Communications of The ACM*, 40(3):56–58, 1997.

[22] Scipy.org. ⟨`http://www.scipy.org/`⟩.

[23] B. Smyth and L. McGinty. The power of suggestion. In *Proc. of the 18th Int'l Joint Conf. on Artificial Intelligence*, pages 127–132, 2003.

[24] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[25] I. Žliobaitė, F. Kamiran, and T. Calders. Handling conditional discrimination. In *Proc. of the 11th IEEE Int'l Conf. on Data Mining*, 2011.

[26] S. Watanabe. *Knowing and Guessing – Quantitative Study of Inference and Information*. John Wiley & Sons, 1969.

[27] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proc. of the 14th Int'l Conf. on World Wide Web*, pages 22–32, 2005.

The Effect of Sensitivity Analysis on the Usage of Recommender Systems

Martina Maida
Vienna University of Economics and Business
Augasse 2-6
1090 Vienna
martina.maida@wu.ac.at

Konradin Maier
Vienna University of Economics and Business
Augasse 2-6
1090 Vienna
konradin.maier@wu.ac.at

Nikolaus Obwegeser
Vienna University of Economics and Business
Augasse 2-6
1090 Vienna
nikolaus.obwegeser@wu.ac.at

Volker Stix
Vienna University of Economics and Business
Augasse 2-6
1090 Vienna
volker.stix@wu.ac.at

ABSTRACT

Recommender systems have become a valuable tool for successful e-commerce. The quality of their recommendations depends heavily on how precisely consumers are able to state their preferences. However, empirical evidence has shown that the preference construction process is highly affected by uncertainties. This has a negative impact on the robustness of recommendations. If users perceive a lack of accuracy in the recommendation of recommender systems, this reduces their confidence in the recommendation generating process. This in turn negatively influences the adoption of recommender systems. We argue in this paper that sensitivity analysis is able to overcome this problem. Although sensitivity analysis has already been well studied, it was ignored to a large extent in the field of recommender systems. To close this gap, we propose a research model that shows how a sensitivity analysis and the presence of uncertainties influence decision confidence and the intention to use recommender systems.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]; J.4 [**Social and Behavioral Sciences**]

General Terms

Theory, Human Factors

Keywords

Recommender systems, sensitivity analysis, uncertainties in preference construction, technology acceptance

Paper presented at the 2012 Decisions@RecSys workshop in conjunction with the 6th ACM conference on Recommender Systems. Copyright ©2012 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

1. INTRODUCTION

Recommender systems (RS) have become an important tool for successful e-commerce. They help consumers in e-commerce settings to overcome the problem of information overload, which they often face due to the vast amount of available products and of product-related information. From a consumers-perspective, the main task of RS is to support finding the right product. Independent from technical considerations, all RS have in common that they require information about their users in order to provide personalized recommendations. This information is basically the consumers' preferences which serve as input for the recommendation-generating algorithm [24]. Thus, the users' preferences are clearly of high importance for the quality of the RS' output and the more precise the preferences correspond to the user's "real" needs, the more accurate will be the recommendation of the system.

The problem we want to address here is that the preferences of consumers as well as their measurement are subject to irreducible arbitrariness [12], which potentially has a negative impact on the quality of a RS's recommendation and on the adoption of RS. To overcome this problem, we propose to integrate sensitivity analysis into RS. The remainder of this paper is structured as follows. The next Section describes the uncertainties related to the measurement of preferences and the implications for RS design. Section 3 provides a short overview of SA methods and possible ways to address uncertainties as well as similar problems of supporting consumers via RS. We will propose a research model in Section 4 and hypothesize how SA and uncertainties in the process of generating recommendations are related to RS usage. The planned methodology for testing our hypotheses is presented in Section 5. Finally, we provide a short discussion of our model and present further research opportunities in Section 6.

2. UNCERTAINTY AND RECOMMENDER SYSTEMS

Humans often face decisions which have to be made based on beliefs regarding the likelihood of uncertain events like future prices of goods or the durability of a product [21].

Here, uncertainty refers to a state of incomplete knowledge, which is usually rooted in either the individual's lack of information or in his limited resources to rationally process the available information [4, 18].

The latter source of uncertainty - limited information processing capabilities - is the rationale underlying the idea to support consumers in making their decisions by providing personalized recommendations. In this sense, it is the function of RS to mitigate the information overload which consumers often face in e-commerce settings [16]. As research in RS deals with bounded rational consumers, it has to acknowledge that consumers face uncertainties while making their purchase decisions, even if they are supported by a RS. The origins of uncertainty in a RS-facilitated purchase decision can be manifold. For example, a consumer might ask himself whether the model underlying the RS is indeed appropriate to support him or whether the complex calculations underlying a recommendation have been solved accurately or in a more heuristic way [4]. Another important source of uncertainty is the consumer. Often, it is assumed that decision makers have stable and coherent preferences and sometimes it is even supposed that they accurately know these preferences [9]. However, there is vast empirical evidence that these assumptions do not model real world decision makers very well. For example, it is commonly known that the answers of a decision maker who is requested to explicitly state his preferences are at least partly dependent on the framing of the questions and on what response is expected [22]. These and other empirically observed deviations from rationality led to the notion that humans do not have well-defined preferences which can be elicited but that we construct preferences on the spot, usually by applying some kind of heuristic information processing strategy. Consequently, our preferences are "labile, inconsistent, subject to factors we are unaware of, and not always in our own best interests" [9, p.2].

For the effort to support consumers with the help of RS such instable preferences pose a serious problem. RS try to support consumers by providing personalized recommendations based on the consumer's preferences. Independent of how the RS measures the preferences of the consumer (either explicitly by asking the consumer or implicitly by observing his behavior), the ad-hoc construction of preferences implies that RS have to deal with an uncertain information base to make recommendations (cf. [4]), which might lead to inaccurate and therefore unhelpful recommendations. Moreover, a consumer who faces a recommendation of a RS might *perceive* a state of uncertainty regarding the recommendation's quality because the choice of the recommendation-generating algorithm, its inputs (the preferences) as well as its computation are afflicted with uncertainties. The work of Lu et al. [10] shows that a major reason for the rejection of decision support technologies is that humans are skeptical whether the respective technology is indeed able to accurately model their preferences. In other words, the uncertainties related to technologically derived recommendations might hamper the adoption of RS. In order to avoid these problems, RS have to address the uncertainties related to the generation of recommendations. Here, we propose to incorporate SA into RS to overcome this challenge.

3. SENSITIVITY ANALYSIS

Sensitivity analysis is a widely used tool in various disciplines, like in chemical engineering, operations research or management science [20]. According to French [5], a common definition of SA involves the variation of input variables to examine their effect on the output variables. In the case of RS, inputs refer to preferences of consumers and output means the recommendation of the system. Thus, SA is a valuable tool for detecting uncertainties in inputs, verification and validation of models as well as demonstrating the robustness of outputs. Definition and purpose however vary depending on the field of application [15]. Furthermore, there are different SA methods. They are classified e.g. in mathematical, statistical and graphical methods [6] or in local and global SA methods depending if the input variables are varied over a reduced range of value or over the whole domain [15]. Both classes allow to vary "one factor at a time" (OAT) or several variables simultaneously (VIC - variation in combination). Some researchers (e.g. [17]) argue that a variance-based, global SA with VIC is especially useful for comparing input variables and identifying uncertainties.

Although SA is in general a well-studied topic, it is ignored to a large extent in the field of RS. Papers that treat SA as tool for decision support systems are typically from the field of multi-criteria decision making. They explain for instance how SA demonstrates robust solutions or illustrates the impact of input variations [13]. A reason why SA should be integrated in decision support systems is that it addresses certain drawbacks, like a possible lack of transparency. By considering RS, this would mean that consumers do not receive the possibility to understand why a particular product was recommended. Thus, consumers are not able to detect uncertainties that were introduced during preference elicitation. As argued by [19, p. 831] "(...) users are not just looking for blind recommendations from a system, but are also looking for a justification of the system's choice.". A possibility to provide justifications are explanation facilities. An approach that was found in literature is to regard SA as being similar to an explanation facility [14]. It facilitates the involvement of users and increases transparency of the recommendation generating process [8]. An integrated SA permits users to interact with the system such that they are able to explore possible variations of the inputs and see how their changes influence the robustness of the recommendation. A SA is therefore especially important when uncertainties in the inputs are present. In contrast to the various types of explanation facilities, it is based on formal sciences and is thus capable of providing objective explanations.

4. RESEARCH MODEL AND HYPOTHESES DEVELOPMENT

Based on the descriptions of the problem of uncertainties and the characteristics of SA we will derive a research model for RS usage in this Section. In order to understand how SA is related to the adoption of RS, we integrate *sensitivity analysis*, *perceived uncertainty* and *decision confidence* in a common model of RS usage. The definitions of these concepts are given in Table 1. Our model builds on technology acceptance research and its most prominent model, the technology acceptance model (TAM) [2]. Figure 1 illustrates the proposed model. *Sensitivity analysis* represents the design feature of interest, *decision confidence* and *perceived uncertainty* are used to describe the link between the design feature and RS use in detail. The following paragraphs

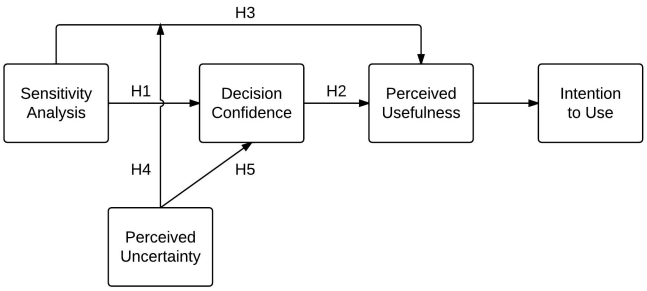


Figure 1: Proposed research model

separately discuss each proposition of our model.

Basically, a SA can lead to two different results: Depending on inputs and model parameters, it will either confirm or disprove the robustness of the recommendations provided by the RS. Though we acknowledge that the output of a SA depends on the specific situation and that the concrete outcome of the SA is likely to influence the user's perceptions, we argue that there is also an effect which is independent from such contingencies (see also Section 6). SA helps users to filter out those recommendations which are robust to uncertainties and which thereby represent good choices independent from changes in the inputs [12]. Therefore, we hypothesize that

> **H1:** *Sensitivity analysis will increase users' decision confidence.*

The only task which RS perform is to search and suggest decision alternatives on behalf of their users. If a user is not sure whether a RS provides recommendations which match his needs or not, the only reason to use a RS vanishes. Therefore, we hypothesize that

> **H2:** *Decision confidence will positively affect perceived usefulness of recommender systems.*

SA is a tool which demonstrates how the output varies when inputs are changed. This enables user not only to analyze different scenarios and to search for robust recommendations but also to learn about the RS and how it generates recommendations. In this function, SA might be directly related to perceived usefulness of the RS regardless of its impact on decision confidence and independent from whether it confirms the robustness of the recommendation or not. Based on this argument and on the experiences of Payne et al. [12] that user perceive SA as a valuable tool, we hypothesize that

> **H3:** *Sensitivity analysis will positively influence perceived usefulness of recommender systems.*

We argue that this relationship is moderated by the degree of perceived uncertainty: Consider a user who does not perceive any uncertainty related to the output of a RS. For such a user a SA is of little to no value. But the more the user perceives that the recommendation generating process is prone to uncertainties, the more useful is a feature which allows to explore the impact of the uncertainties on the outcomes. Therefore, we hypothesize that

> **H4:** *Perceived uncertainty will moderate the influence of sensitivity analysis on perceived usefulness of recommender systems.*

Table 1: Definitions of Constructs

Construct	Definition
Sensitivity Analysis	A RS feature which allows a user to analyze how a recommendation (output) changes if the preferences (inputs) are varied [5]
Decision Confidence	The user's beliefs that the recommendation matches his preferences [7]
Perceived Uncertainty	The user's subjective probability assessment of any presence of inaccuracy in the recommendation generating process [4]
Perceived Usefulness	The user's perceptions of the utility of the RS [24]
Intention to Use	The user's subjective probability of adopting the RS [3]

The relationship between perceived uncertainty and decision confidence is similar to H4. If users perceive that a recommendation is based on an uncertain information base or if they are not sure about the appropriateness of the recommendation generating algorithm, they are likely not confident about the quality of the recommendation. Therefore, we hypothesize that

> **H5:** *Perceived uncertainty will negatively influence decision confidence.*

5. PROPOSED METHODOLOGY

We will conduct a laboratory experiment to test our hypotheses. We will use a 2 x 2 full factorial design with SA and perceived uncertainty as independent variables. Participants will be asked to use a RS for online shopping which explicitly demands from users to make trade-offs in preference construction. They will be randomly assigned to a treatment group and a control group which allows us to manipulate SA and perceived uncertainty. We will choose purchase decisions with low/high familiarity to induce high/low levels of perceived uncertainty. After finishing the shopping task, questionnaires will be delivered to the participants to assess the proposed relationships.

Before we are actually able to conduct the experiment, we will develop new measures for the constructs perceived uncertainty and decision confidence by adopting the method of Moore and Benbasat [11] for instrument development. The validity and reliability of the items will be tested by a factor analysis in a pilot test. Items for the remaining constructs will be taken from already validated scales, for instance from Davis [2] for perceived usefulness.

To test our experimental design, we will conduct a t-test in order to check the manipulation of perceived uncertainty via familiarity of the purchase task. For testing our hypotheses we will use structural equation modeling (SEM). As our study is the first one regarding the impact of SA and uncertainty on RS usage, it has an exploratory character. To manage the risks associated with exploratory research, we will keep the sample size rather low (about 10 participants per indicator [1]). To deal with the small sample size and the

exploratory character of our research, we will use a partial least squares approach (component-based SEM) [23].

6. DISCUSSION AND CONCLUSIONS

Based on a literature review, we have argued that the process of generating recommendations for e-commerce users involves uncertainties, especially regarding the measurement of preferences, which might lead to users who feel insecure about the quality of a RS's recommendations. Moreover, we hypothesized that if users do not feel confident about a RS's recommendations, they will not perceive RS as useful and thus are less likely to adopt the RS. We proposed to incorporate SA into RS to overcome the problems associated with uncertainties. SA is a tool which enables users to explore how changes in the inputs of the recommendation generating process (the users' preferences) are related to changes in the output of the process (the recommendations). SA can be used to check the robustness of recommendations which should help users to build confidence in the system's advice and the decision. Finally, we proposed a conceptual model and corresponding hypotheses of how uncertainties, decision confidence and SA are related to the adoption of RS.

As outlined in Section 5 our next step is the empirical testing of the proposed model by conducting a laboratory experiment. Further research opportunities include theoretical work on how SA can be incorporated into the various forms of RS, not only on computational level but also on the level of user interface design and how the outcomes of SA are related to user perceptions.

7. ACKNOWLEDGMENTS

This research has been funded by the Austrian Science Fund (FWF): project number TRP 111-G11.

8. REFERENCES

[1] J. Christopher Westland. Lower bounds on sample size in structural equation modeling. *Electronic Commerce Research and Applications*, 9(6):476–487, 2010.

[2] F. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340, 1989.

[3] M. Fishbein and I. Ajzen. *Belief, attitude, intention, and behavior: an introduction to theory and research.* Addison Wesley Publishing Company, 1975.

[4] S. French. Uncertainty and imprecision: modelling and analysis. *The Journal of the Operational Research Society*, 46(1):70–79, 1995.

[5] S. French. Modelling, making inferences and making decisions: the roles of sensitivity analysis. *Top*, 11(2):229–251, 2003.

[6] H. Frey and S. Patil. Identification and review of sensitivity analysis methods. *Risk Analysis*, 22(3):553–578, 2002.

[7] D. Griffin and A. Tversky. The weighing of evidence and the determinants of confidence. *Cognitive Psychology*, 24(3):411–435, 1992.

[8] J. Herlocker, J. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

[9] S. Lichtenstein and P. Slovic. The construction of preference: an overview. In *The construction of preference, Lichtenstein, S. and Slovic, P. (Eds.)*, pages 1–40. Cambridge University Press, 2006.

[10] H. Lu, H. Yu, and S. Lu. The effects of cognitive style and model type on dss acceptance: An empirical study. *European Journal of Operational Research*, 131(3):649–663, 2001.

[11] G. Moore and I. Benbasat. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information systems research*, 2(3):192–222, 1991.

[12] J. Payne, J. Bettman, and D. Schkade. Measuring constructed preferences: Towards a building code. *Journal of Risk and Uncertainty*, 19(1–3):243–270, 1999.

[13] L. Proll, A. Salhi, and D. Rios Insua. Improving an optimization-based framework for sensitivity analysis in multi-criteria decision-making. *Journal of Multi-Criteria Decision Analysis*, 10(1):1–9, 2001.

[14] W. Raskob, F. Gering, and V. Bertsch. Approaches to visualisation of uncertainties to decision makers in an operational decision support system. In *Proceedings of the 6th International ISCRAM Conference*, page http://www.iscram.org/ISCRAM2009/papers/, 2009.

[15] A. Saltelli, K. Chan, and E. Scott. *Sensitivity analysis.* John Wiley, New York, 2000.

[16] J. Schafer, J. Konstan, and J. Riedl. E-commerce recommendation applications. *Data mining and knowledge discovery*, 5(1):115–153, 2001.

[17] K. Siebertz, D. Van Bebber, and T. Hochkirchen. *Statistische Versuchsplanung: Design of Experiments (DoE).* Springer Verlag, 2010.

[18] H. Simon. *A behavioural model of rational choice, In: Models of Man: Social and Rational.* John Wiley, New York, 1957.

[19] R. Sinha and K. Swearingen. The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 830–831. ACM, 2002.

[20] E. Triantaphyllou and A. Sánchez. A sensitivity analysis approach for some deterministic multi-criteria decision-making methods*. *Decision Sciences*, 28(1):151–194, 1997.

[21] A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131, 1974.

[22] A. Tversky and D. Kahneman. The framing of decisions and the psychology of choice. *Science*, 211(4481):453–458, 1981.

[23] N. Urbach and F. Ahlemann. Structural equation modeling in information systems research using partial least squares. *Journal of Information Technology Theory and Application*, 11(2):5–40, 2010.

[24] B. Xiao and I. Benbasat. E-commerce product recommendation agents: Use, characteristics, and impact. *MIS Quarterly*, 31(1):137–209, 2007.

Recommending Personalized Query Revisions

Henry Blanco
Faculty of Computer Science
Free University of Bolzano
Bolzano, Italy.
Center of Medical Biophysics
University of Oriente
Santiago de Cuba, Cuba.

Francesco Ricci
Faculty of Computer Science
Free University of Bolzano
Bolzano, Italy.
fricci@unibz.it

Derek Bridge
Department of Computer Science
University College Cork
Cork, Ireland.
d.bridge@cs.ucc.ie

ABSTRACT

Observing the queries selected by a user, among those suggested by a recommender system, one can infer constraints on the user's utility function, and can avoid suggesting queries that retrieve products with an inferior utility, i.e., dominated queries. In this paper we propose a new efficient technique for the computation of dominated queries. It relies on the system's assumption that the number of possible profiles (or utility functions), of the users it may interact with, is finite. We show that making query suggestions is simplified, and the number of suggestions is strongly reduced. We also found that even if the system is not contemplating the true user profile, among the above mentioned finite set of profiles, its performance is still very close to the optimal one.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering*

General Terms

Experimentation, Theory.

Keywords

Recommender system, conversational system, user model.

1. INTRODUCTION

Conversational recommender systems offer flexible support to users as they browse a product catalogue, and help them to better understand and elicit their preferences. Instead of requiring users to specify their preferences at the outset, these are acquired and revised over a series of interaction steps. At each step the system makes some recommendations to the user, or invites her to indicate further preferences, e.g., by critiquing a recommendation [6].

In [3, 9] the authors introduce and evaluate a new conversational technique for helping the users to select items of largest utility to the user. In order to accomplish this goal, when a user is querying a product catalogue the proposed technique suggests to the user new queries that: a) extend the user's current query, and b) retrieve products with higher utility. For example the user of a hotel catalogue may have submitted the following query: "I want an hotel with AC and parking". The system, rather than retrieving immediately the products that satisfy this query, hypothesizes that the user may have also other needs and makes recommendations by suggesting queries that are revisions of the original query. These new queries may add one or more additional features to the query, e.g., the system may say: "are you interested also in a sauna?". Products with more features, if available, will surely increase (or, at least, not decrease) the user utility. But not all features are equally important for the user. So, the system's goal is to make "informed" suggestions, i.e., to suggest those features that are likely to produce the largest increase to the user's utility. In fact, observing the user's previously submitted queries, the system can deduce that certain features are more important than others, i.e., it can infer constraints on the user's utility function, even without knowing that function.

The major limitations of the previous work on this proposed technique were: a) a limited number of query editing operations, i.e., the system could suggest only two types of new queries to the user (add a feature and trade one feature for two), b) a computationally expensive method for computing the next best queries (undominated queries), c) a long list of query suggestions could be possibly presented to the user, making it hard for her to evaluate them and select her preferred one. In this paper we propose a new effective technique for the computation of the dominated queries, i.e., the queries that should not be suggested to the user because the system can deduce that they have a lower utility. The proposed technique relies on the system assumption that the set of profiles (or utility functions) of the users it may interact with is finite. This is a meaningful assumption as not all the possible profiles are likely to ever be observed in practice, and users tend to have similar profiles. We show that the computation of the query suggestions is simplified, and more importantly, the number of queries that are suggested at each conversational step is greatly reduced. We also show that the query suggestions can be further filtered by estimating the utility of each query suggestion using those profiles that are compatible with the queries previously selected by the user. The proposed approach has also another advantage, it enables a system designer to freely select the types

Paper presented at the 2012 Decisions@RecSys workshop in conjunction with the 6th ACM conference on Recommender Systems. Copyright ©2012 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

of query editing operations that he or she would like to use to generate new query suggestions to the user.

We also show that even if the system is not contemplating the true user profile, among the above mentioned finite set of profiles, its performance is still very close to the optimal one, i.e., at the end of the dialogue the user can select the best products, given her true profile and the available products in the data set. Hence the finally recommended items are close to the optimal ones. In fact, we show in this paper that progressively expanding the number of profiles contemplated by the system one can increase the utility of the final recommended products, and with a large number of contemplated profiles the recommended products have a utility that is not practically distinguishable from that of the best products.

The rest of the paper is structured as follows. The query language used in this approach is described in Section 2. Section 3 describes our model for representing user preferences. Section 4 explains the concept of "dominated query" and our query suggestion method. The experimental design is shown in Section 5. Results and discussion are reported in Section 6. Finally the related work and conclusions are given in Sections 7 and 8 respectively.

2. QUERY LANGUAGE

In our model a product p is represented by an n-dimensional Boolean feature vector $p = (p_1, \ldots, p_n)$. $p_i = 1$ means that the i-th feature (e.g., Air Conditioning) is present in the product, whereas $p_i = 0$ means that p does not have feature i. A catalogue is a set of products $\{p^{(1)}, \ldots, p^{(k)}\}$. The Boolean features could be keywords or tags found in the product description, and searching for products with these features can be viewed as kind of facet search.

Queries are represented similarly as Boolean vectors: $q = (q_1, \ldots, q_n)$. $q_i = 1$ means that the user is interested in products that have the i-th feature. On the other hand $q_i = 0$ does not mean that the user is not interested in products with that feature, but simply that she has not yet declared her interest on it. A query is said to be *satisfiable* if there exists a product in the catalogue such that all the features expressed in the query as desired ($q_i = 1$) are present in that product. For example if the product $p = (1, 1, 0, 1, 0)$ is present in the catalog then query $q = (0, 1, 0, 1, 0)$ is satisfiable.

We are considering a scenario where the user is advised about how to refine her queries. Moreover, we assume that the system GUI offers to the user a limited number of easily understood editing operations (as in critiquing-based approaches). In the following we list the query editing operations that, in this paper, we assume the user can make when revising the current query. But we observe (as will be clear in the ensuing description) that the proposed approach is not constrained by this particular choice.

- $add_1(q, i)$, $i \in idx0(q)$;
- $trade_{1,2}(q, i, j, k)$, $i \in idx1(q)$ and $j, k \in idx0(q)$;
- $add_2(q, i, j)$, $i, j \in idx0(q)$;
- $trade_{1,3}(q, i, j, k, t)$, $i \in idx1(q)$ and $j, k, t \in idx0(q)$.

Here $idx0(q)$ and $idx1(q)$ are the set of indexes with value 0 and 1 in q respectively. The first two operations generate a new query by adding to the current query a request for one additional feature. For example, in $(1, 1, 0, 0, 1) = add_1((1, 1, 0, 0, 0), 5)$ the query $q = (1, 1, 0, 0, 0)$ (where the first two features are requested) is extended by requesting also the fifth feature. The second operation (trade one feature present for two not present) generates a new query by discarding a feature, the i-th, in favor of two new ones, the j-th and k-th features. For example, $(0, 1, 0, 1, 1) = trade_{1,2}((1, 1, 0, 0, 0), 1, 4, 5)$.

The last two operations extend a query with two additional features. For example, given the query $q = (1, 1, 0, 0, 0)$, the fourth and fifth features can be requested in the new query generated by the operation $add_2(q, 4, 5) = (1, 1, 0, 1, 1)$. The second "trade" operation (trade one feature present for three not present) generates a new query by discarding the i-th feature, but now in favor of three new ones, the j-th, the k-th and t-th features. For example, $(0, 1, 1, 1, 1) = trade_{1,3}((1, 1, 0, 0, 0), 1, 3, 4, 5)$.

Using the above-mentioned operators the system can generate a set of next queries and ask the user to select her preferred one, and this step can be repeated several times (see Section 6.1 for an example of such an interaction). However, the goal of the proposed system is not to suggest all these possible next queries, as a standard "query by example" interface might, but only those that would retrieve products with the largest utility for the user. Hence, the goal of the proposed system is to make inferences on the true user utility function, and remove from the suggestions it makes at every step those queries that appear to the system to have an inferior utility. This reasoning process is clarified in the following sections.

3. USER UTILITY FUNCTION

A user's utility function, also called her user profile, is represented here as a vector of weights: $w = (w_1, \ldots, w_n)$, $0 \leq w_i \leq 1$. w_i is the importance that the user assigns to the i-th feature of a product. So if $w_i = 0$, then the user has no desire for the i-th feature. If $w_i > w_j$, then the i-th feature is preferred to the j-th one. If $w_i \geq w_j$ then the i-th feature is at least as desired as the j-th one. If $w_i = w_j$, $i \neq j$ then the user is indifferent between these two features. The user's utility for a particular product $p = (p_1, \ldots, p_n)$ is given by the following:

$$Utility_w(p) = \sum_{i=1}^{n} w_i p_i \quad (1)$$

A product p with a higher utility than another product p' is always assumed to be preferred by the user, i.e., we assume that users are rational. A user may have any of the possible utility functions that can be defined by varying the feature weights w_i. So, the set of all possible utility functions is infinite. But observing the queries selected by the user among those suggested by the system the system can infer constraints on the definition of the true user utility function. Generally speaking, features that are present in a query that the user selects can be inferred to be more desirable for that user than features that are present in the alternative queries that the user could have tried but did not select.

More precisely let us assume that the system recommends to the user a set of new queries, which we will call the AdviceSet. The queries in the AdviceSet will, in general, be a subset of the queries that can be generated by applying the query editing operations described in the previous section to the query that was selected by the user at the previous

interaction step. When the user selects one of these recommended new queries, as the new best query for her, the system can deduce that the utility of the one she selects is greater than or equal to the utility of the other queries that were included in the AdviceSet. If we define $Utility_w(q)$, the utility of query q for a user with profile w, as the utility of a product p with the same definition as q, i.e., $q = p$, then, if the user selects $q_s \in AdviceSet$, we can infer that:

$$Utility_w(q_s) \geq Utility_w(q),\ \forall q \in AdviceSet. \quad (2)$$

For example: Let's assume that the previous query selected by the user is $q_0 = (0, 0, 1, 1, 0, 0, 0)$, i.e., there are seven features in this data set and the user would like to retrieve products having the third and fourth feature. Assume that the system suggests that the user edits the current query and specifically recommends that she select one of the following four queries:

$$\begin{aligned} AdviceSet = \ & \{(1, 0, 1, 1, 0, 0, 0), \\ & (0, 1, 1, 1, 0, 0, 1), \\ & (0, 0, 1, 1, 1, 0, 0), \\ & (0, 0, 1, 0, 0, 1, 1)\} \end{aligned}$$

Let us further assume that the query that the user selects from these recommended ones is $q_s = (0, 0, 1, 1, 1, 0, 0)$, i.e., she is interested in products that additionally have the 5-th feature. Then, the inferred constraints, based on not choosing other members of the AdviceSet, are:

1. $w_5 >= w_1$
2. $w_5 >= w_2 + w_7$
3. $w_4 + w_5 >= w_6 + w_7$

We must also explain what constraints on the true user profile w can be deduced when the user issues the very first query in any interaction. In this case, if q is the initial query, the advisor will infer that $w_i \geq w_j$, $\forall i \in idx1(q)$ and $\forall j \in idx0(q)$, unless q_s, which is identical to q except that its i-th feature is set to 0 and its j-th feature is set to 1, is unsatisfiable. This means that features requested in the initial query are at least as desired as features not initially requested. But, the system must "play safe". In the case where q_s, identical to q but with the i-th feature set to 0 and the j-th feature set to 1, is unsatisfiable, it should not deduce a constraint of the type $w_i \geq w_j$. This is because there is the possibility that the user already knew this query to be unsatisfiable and for this reason she did not try it as her initial query, even though she preferred it. A longer discussion of this "play safe" rule is given in [3].

4. THE QUERY ADVISOR

The advisor is the recommender system in charge of suggesting to the user how to extend the current query to obtain better products, i.e., it generates the AdviceSet. The true user's preferences, in her profile, are not known by the advisor. Moreover, we assume that the advisor does not explicitly ask the user for her preferences. Nevertheless, right after the user's first query, the advisor will generate a set of candidate queries and will recommend only the undominated candidates, i.e. those with a utility that cannot be proved to be inferior to one of the other candidates. Each time the user chooses one of the recommended queries, the advisor makes new recommendations. It does this repeatedly until the user is happy with her current query or no additional suggestions can be made by the system.

At each interaction step, the advisor accumulates constraints on the true user utility function (as described in Section 3). We denote this set of constraints by Φ. Moreover, given a set of next possible queries $C = \{q^{(1)}, \ldots, q^{(k)}\}$, i.e., those that can be generated by applying the operations described in Section 2, and that are satisfiable, the advisor will not suggest queries that have a lower utility than another one: these queries are called here "dominated". A query $q \in C$ is *dominated* if there exists another query $q' \in C$ such that for all the possible weight vectors that are compatible with the set of constraints Φ this relation holds: $Utility_w(q') > Utility_w(q)$. A weight vector w is said to be *compatible* with the set of constraints in Φ if and only if all the constraints in Φ are satisfied when the variables $w_1, \ldots, w_n$ take the values specified in w.

Removing the dominated queries is meaningful because their utility is lower than the utility of another candidate query for all the possible user utility functions that are compatible with the preferences that have been induced by observing the user's behavior.

In our previous work, the problem of finding dominated queries was cast as a linear programming problem, allowing an infinite number of user profiles to be considered. The problems with this are discussed in Section 7. In this paper we assume that the set of user profiles contemplated by the system is finite. Initially, at the beginning of the interaction with a user, the set of all the possible utility functions or user profiles is $P = \{w^{(1)}, \ldots, w^{(m)}\}$. We will consider in the experiments sets of user profiles ranging from some dozens to thousands.

With this finite assumption, having the set of deduced constraints Φ we can prune from the set P the "incompatible profiles", i.e., those not satisfying the constraints in Φ. Then, the computation of the undominated queries proceeds as follow. Let's assume that the set of user profiles compatible with the accumulated constraints is $P' = \{w^{(1)}, \ldots, w^{(t)}\} \subset P$ and $C = \{q^{(1)}, \ldots, q^{(k)}\}$ is the set of next possible queries, i.e., queries that are satisfiable and are generated from the last issued query of the user by the query editing operations. Then the AdviceSet is given by the following method:

1. A query $q \in C$ is labelled as dominated if and only if there exists another query $q' \in C$, $q' \neq q$, such that $\forall w \in P'$, $Utility_w(q') > Utility_w(q)$. , i.e., $\sum_{i=1}^{n} w_i q'_i > \sum_{i=1}^{n} w_i q_i$.
2. Build the AdviceSet (undominated queries) by removing from C the dominated queries.

Example. Assume that $\Phi = \{w_1 \geq w_3, w_2 + w_3 \geq w_4\}$, $P' = \{w^{(1)}, w^{(2)}, w^{(3)}\}$ and $C = \{q^{(1)}, q^{(2)}, q^{(3)}, q^{(4)}\}$, $w^{(1)} = (0.35, 0.1, 0.25, 0.3)$, $w^{(2)} = (0.1, 0.35, 0.3, 0.25)$, $w^{(3)} = (0.3, 0.35, 0.1, 0.25)$, $q^{(1)} = (1, 1, 0, 1)$, $q^{(2)} = (1, 0, 1, 1)$, $q^{(3)} = (0, 1, 1, 1)$, $q^{(4)} = (1, 1, 1, 0)$.

In this example the profiles $w^{(1)}$ and $w^{(3)}$ satisfy the constraints in Φ. While, $w^{(2)}$ is an "incompatible profile", since $w_1^{(2)} < w_3^{(2)}$, and must be pruned from P'. Table 1 shows the query utilities for these two compatible profiles. $q^{(1)}$ has a higher utility than $q^{(3)}$ and $q^{(4)}$ for every profile in P', thus $q^{(3)}$ and $q^{(4)}$ are dominated by $q^{(1)}$. These dominated

Table 1: Query utilities for the profiles $w^{(1)}$ and $w^{(3)}$.

	$\mathbf{q^{(1)}}$	$\mathbf{q^{(2)}}$	$\mathbf{q^{(3)}}$	$\mathbf{q^{(4)}}$
$\mathbf{w^{(1)}}$	0.75	0.9	0.65	0.7
$\mathbf{w^{(3)}}$	0.9	0.65	0.7	0.75

1. $\Phi = \emptyset$, P =set of profiles, AdviceSet = empty set
2. **do** {
3. Present the AdviceSet to the user.
4. sq = initial query or one in the AdviceSet;
5. Infer constraints analyzing sq, and add them to Φ;
6. Remove incompatible profiles from P;
7. Compute candidate queries;
8. Remove dominated queries from candidate ones and generate AdviceSet;
9. *(optional)* Filter the AdviceSet;
10. } **while** ((AdviceSet $\neq$ null) and (user wants advice))

Figure 1: Interaction process

queries must be removed from the set C and not included in the AdviceSet. Note that the remaining queries $q^{(1)}$ and $q^{(2)}$ do not dominate each other, thus they represent meaningful next queries that the advisor can recommend to the user.

The full algorithm for query suggestions is described in Figure 1. At the first step there are no query suggestions, and the user is free to enter the first query. Then, the advisor infers the constraints to be added to Φ according to the rules mentioned in Section 3. The advisor then removes the user profiles that do not satisfy these constraints. Afterwards, the set of candidate queries is generated from the current query, by applying the operators mentioned in Section 2 and discarding any queries that are not satisfiable. Subsequently, the advisor builds the AdviceSet by removing the dominated queries, and optionally filters the AdviceSet to keep it small. The filtering strategy that we have applied will be presented in the next section. Finally, the advisor recommends the remaining queries to the user as potential next 'moves'. If the AdviceSet is not empty, and the user selects one from this advice set, then the selected query becomes the current query and the process is repeated. If the user does not want further advice then the system will display the products that satisfy the last query selected by the user.

5. EXPERIMENTS DESIGN

We performed several experiments by simulating interactions between a virtual user and the advisor according to the algorithm described in the previous section. For each experiment we varied the following independent variables: product database, number of predefined user profiles, and whether the undominated queries were filtered or not in order to reduce the number of suggestions in the AdviceSet (step 9 of the algorithm). We measured: the average number of queries issued per dialogue (interaction length), the average size of the AdviceSet(number of queries suggestions at each step), the utility shortfall, and the Jaccard similarity between the last selected query and the optimal one. The utility shortfall (or "regret") is the difference between the utility of the best product available in the data set, i.e., the one with the highest utility for the user, and the utility of the products satisfying the last query selected by the user. This measure indicates if the advisor's suggestions do converge on the best product according to the true utility function, hence if the final product recommendations are optimal. Moreover, in order to understand how many features differ between the user's best product and the products satisfying the last query considered by the user, which in a real scenario would be the products actually shown to the user, we measured their Jaccard similarity. This is the ratio of the number of features common to the best product and the last query, over the number of features in their union. In practice, the utility shortfall can be very small (if the features that differ in the best product and in the last query have small weights in the user's utility function), but the Jaccard similarity could still be far from 1.0.

Table 2: Product databases. (Dist. Hotels = Distinct Hotels)

Name	Features	Hotels	Dist. Hotels
Marriot-NY	9	81	36
Cork	10	21	15
Trentino-10	10	4056	133

Three different product databases were used, each one describing real hotels by their amenities expressed as Boolean features. Details of the data set are given in the Table 2; here a hotel may have the same description in terms of features as another; that's why the number of distinct hotels is smaller. Moreover, we considered for each experiment four different sizes of the set of predefined user profiles: small (25 profiles), medium (250 profiles), large (2500 profiles) and very large (25000 profiles). We wanted to measure the effect of the size of the profiles set on the user-advisor interaction length, and on the size of the advice set.

In each experiment a set of predefined user profiles is created by first generating one totally random initial user profile (weights vector), sampling each random feature weight from a uniform distribution in [0,1], and then normalizing the user profile vector so that the sum of the weights is 1. Then, the other profiles are created by random permutations of the feature weights of the initial user profile. Note that with 10 features there are $3.6 \times 10^6 \sim 10!$ possible user profiles.

For step 9 of the algorithm, i.e., the optional filtering of the query suggestions in the advice set to produce an AdviceSet that has at most a small number of suggested queries (5 in our case), we used one strategy. We considered the strategy that selects the top K queries in the AdviceSet, with the largest expected utility. The expected utility of each query in the AdviceSet is computed by averaging the utility of the query for all the profiles compatible with the inferred constraints. This approach assumes that the compatible profiles have equal probability to be the true profile of the user.

In addition to the user profiles contemplated by the advisor, in each simulated interaction we randomly generated the true profile of the virtual user and it was not revealed to the advisor. Note that the true virtual user profile is very unlikely to be among the predefined set of advisor user profiles. Moreover, the initial query submitted by the virtual user is created in accordance with her true utility func-

tion; thus, the initial query includes the t most important features for the user ($t = 2$ in our experiments). The advisor's deductions about the true user utility function are based only on the observation of the queries submitted by the user at each interaction step. We also assumed that the virtual user is "Optimizing" [3], that is, one who confines her queries to the advice set provided by the advisor and always tries the query with the highest utility. Twenty-four experiments were performed corresponding to all the combinations of the variables mentioned before (product database, number of user profiles, filtering strategy). In every experiment we ran 100 dialogues between a virtual user and the advisor and then averaged the observed measures.

6. RESULTS AND DISCUSSION

6.1 Example of Simulated Interaction

Before describing the results of the system evaluation we want to illustrate with one example a typical user-advisor interaction. In this example we are considering the Marriott catalogue, and the system is using the utility-based filtering strategy, hence no more than 5 queries will be recommended at each step. Some of the details are in Table 3.

The features, numbered from 0 to 8 are: 0=*Internet access point*, 1=*Restaurant on site*, 2=*Room service*, 3=*Pets allowed*, 4=*Meeting room*, 5=*Airport shuttle*, 6=*Swimming Pool*, 7=*Golf camp*, 8=*Tennis camp*. The five most important features for the simulated user in this example are {0,1,2,5,8}, but there is no hotel with exactly these features in the dataset, and the best available hotel is {0,1,2,3,5}.

The user starts the interaction with the query which, according to her preferences, contains the two most important features: {1,2}. The system infers some initial constraints from the initial query, i.e., that these features are more important that the others not requested (see Section 3), and discards the profiles that do not satisfy these constraints. In this case it discovers that 10 out of 250 initial predefined profiles satisfy the inferred constraints (compatible profiles). These profiles are considered by the system as those potentially containing the true user profile and thus will be examined to make new query suggestions. The system computes the next candidate queries and discards those that are not satisfiable: (58 are candidates) Then the advisor removes those that are dominated, the remaining queries (10 undominated) are ranked by computing their expected utility, and the top 5 are suggested. Note that the queries suggested extend the previous one with extra features. The query selected by the user is {1,2,5,6}, since it is the one that maximizes her utility. At this point the utility shortfall is 0.136 and the Jaccard similarity with the best hotel, {0,1,2,3,5}, is 0.5 (3 common features out of 6 in the union). The system infers 4 new constraints: these constraints state that the utility of the selected query is greater than or equal to the utility of the other queries that were suggested. The number of compatible profiles is now 2, and only 1, that is {0,1,2,5,6}, out of the 7 satisfiable queries, is undominated, and thus suggested to the user. It is interesting to note that the best (and satisfiable) query $\{0,1,2,3,5\} = trade(\{1,2,5,6\},6,1,3)$ was (erroneously) considered by the system to be dominated by the suggested query {0,1,2,5,6}, and therefore not included in the advice set. This results from the fact that the dominated queries are computed using the compatible profiles (2 in this case) not the true user model, which is unknown to the system. These two compatible profiles (erroneously) assign a higher weight to feature 6 (*Swimming Pool*) instead of feature 3 (*Pets allowed*) as it is stated in the true user profile. In the third interaction step the user is forced to select the unique query that is suggested. At this point it is not possible to extend the current query with a satisfiable one any further, the system cannot make new query recommendations, and the interaction ends. The utility shortfall and the Jaccard similarity are 0.0018 and 0.67 respectively. In this example it is clear that reasoning with a finite set of profiles causes some loss in recommendation accuracy, which is compensated by a speed up in system performance and a reduction in the sizes of the AdviceSet compared with the approach introduced in [3] (see discussion later).

6.2 Interaction Length

Table 4 shows the results of our experiments. The query suggestion strategy based on the utility filtering, as well as the baseline approach (not filtering the query suggestions), produce interaction sessions with average length ranging between 2 and 4.

When the size of the user profiles set is small (25 profiles), the interaction length is even shorter, ranging between 2 and 2.6; this is because it is more likely to fall into a situation where no user profile is compatible with the inferred constraints and the system cannot suggest a new query.

In general, the interaction length is dependent on the number of product features and the available products in the data set. Firstly, the higher the number of product features, the longer will be the interaction. This is because the user, at each interaction step, when she is selecting one of the query editing operations, extends the previous query by one or two additional features. Secondly, the smaller the number of products, the more likely the process is to stop, because the current query cannot be further extended without building a failing query. It is important to note that the interaction length is typically low and fairly acceptable for an online application.

6.3 AdviceSet Size

The average size of the advice set ranges between 0 and 12 when no filtering is applied. In this case, inspecting the experiments' log data, we detected that in the initial steps of the user-system interaction, i.e., when the system has poor knowledge about the user preferences, the average number of suggested queries could be as high as 20 (when the system is contemplating a large number of profiles). On the other hand, when the system is filtering the AdviceSet, obviously, the size of the advice set is never greater than $K = 5$. Table 4 shows the average number of queries suggested and, as expected, the filtering strategy (utility-based) produces smaller AdviceSets compared to the not-filtered case. In general, when the size of the set of predefined user profiles is small (25 profiles), the number of query suggestions ranges between 0 and 1.5; this is caused (as we discussed above for the interaction length) by the lack of compatible user profiles, resulting in the difficulty of identifying queries to suggest to the user.

6.4 Utility Shortfall

We expected to observe a higher utility shortfall when filtering the advice set. In fact, in this case, the system

Table 3: An example of the user-system interaction

hotel features:	0	1	2	3	4	5	6	7	8
true user profile:	0.134	0.264	0.188	0.025	7.0e-4	0.141	0.023	0.06	0.164
best hotel:	{0, 1, 2, 3, 5}		**number of initial profiles:**				250		

User	**Advisor**
* Issues the initial query = {1, 2}	* Number of inferred constraints = 11 * Number of compatible profiles = 10 * Number of satisfiable queries = 58 * Undominated queries = 10. Top K=5 suggested: {{1,2,4,6}, {1,2,5,6}, {1,2,3,6}, {1,2,4,5}, {1,2,3,4}}
* Selects the query: {1, 2, 5, 6}	* Number of inferred constraints = 4 * Number of compatible profiles = 2 * Number of satisfiable queries = 7 * Undominated queries = 1: {{0,1,2,5,6}}
* Selects the query: {0, 1, 2, 5, 6}	* No new constraints inferred. * The same number of satisfiable profiles remains. * Number of satisfiable queries = 0. * The system cannot make more query suggestions.
Utility shortfall = 0.0018, **Jaccard Similarity** = 0.667	

Table 4: Averaged values of the observed measures for 100 runs in the 24 experiments performed. (DB = Product Database; # Prof. = Number of predefined user profiles; IL = Interaction Length; AdvSS = AdviceSet Size; USh = Utility Shortfall; JSim = Jaccard Similarity)

DB	**# Prof.**	**Not filtering**				**Utility filtering**			
		IL	**AdvSS**	**USh**	**JSim**	**IL**	**AdvSS**	**USh**	**JSim**
Cork	25	2.57	0.65	0.177	0.575	2.57	0.61	0.177	0.575
	250	3.09	8.32	0.063	0.778	3.6	2.98	0.031	0.895
	2500	3.69	8.43	0.005	0.968	3.81	3.40	0.0	0.991
	25000	3.81	7.69	0.0	1.0	3.84	3.43	0.0	0.993
Marriott	25	2.13	1.12	0.167	0.594	2.13	1.12	0.167	0.594
	250	2.61	8.66	0.033	0.857	2.93	3.33	0.037	0.825
	2500	2.98	7.93	0.0	0.994	3.0	4.25	0.003	0.965
	25000	2.99	7.82	0.0	0.996	3.0	4.22	0.004	0.965
Trentino	25	2.11	0.51	0.324	0.462	2.11	0.5	0.324	0.462
	250	2.95	11.31	0.163	0.626	3.65	2.95	0.080	0.761
	2500	3.65	12.67	0.060	0.797	3.96	3.66	0.018	0.876
	25000	3.99	11.55	0.015	0.890	4.01	3.62	0.012	0.891

may not include in the AdviceSet the best next query, causing, at that step, a loss in the user utility compared with the best query and thus an increase of the utility shortfall. What mitigates this problem is the fact that the system may still suggest the best query at a subsequent interaction step. For instance, if the current query contains two features and the best query contains two additional features, the system, when filtering the suggestions, may not recommend the query using the best of the two missing features at the first step, but it could do it at the next suggestion step.

In general the utility shortfall decreases when the number of user profiles increases. This is true regardless of whether filtering is used or not. When the number of user profiles is small (25 profiles) the utility shortfall values are higher, ranging between 0.2 and 0.3. This is essentially due to the fact that very often the user profiles do not satisfy the constraints inferred by the system. This causes the interruption of the interaction at an early stage. In this case there is not a big difference in the utility shortfall whether filtering query suggestions is used or not, because the size of the advice set never exceeds the threshold $K = 5$.

When the system filters the query suggestions and the user profiles set size is medium (250 profiles) or even larger (2500 profiles), the utility shortfall is very close to that of the not-filtered case. Moreover, in some cases (e.g., Trentino and 2500 profiles) the utility-based strategy may even perform better than the not-filtering approach (0.018 vs. 0.060). This could happen for a very simple reason. When the system suggests fewer queries, the selection of one of these queries by the simulated user causes the system to infer fewer constraints on the user utility function. In fact the system can only deduce that the selected query does not have a lower utility (for the user) than the other suggested queries. Inferring fewer constraints causes the system to eliminate fewer profiles and hence enables the system to make a larger number of interaction steps before arriving at the possibly failing

situation that no profile is compatible with the inferred constraints. This is confirmed by the fact that in these cases (e.g., Trentino and 2500 profiles), where the utility-based approach behaves better than the not-filtering one, the interaction length is on average a bit larger (3.96 vs. 3.65).

In the case where the system is contemplating a large number of user profiles (25000 profiles), filtering the query suggestions has a very small effect. The difference in the utility shortfall with the not-filtering approach is still smaller than 0.0041 (e.g., Marriott 25000, 0.0 vs. 0.004), and there is never a gain in utility. In general the Jaccard similarity between the best hotel and the last selected query increases when the number of predefined profiles increases as well. The Jaccard similarity is higher than 89% when the system is contemplating 25000 profiles. Moreover, this value is better (96%) for the smallest data sets (e.g. Cork and Marriott). These results confirm the previous conclusions on the utility shortfall; it is more likely that better system query suggestions are obtained when the number of predefined user profiles is higher. In fact, it is not important to have many profiles, but rather to have an optimal set of predefined user profiles covering the true user profiles of the subjects accessing the system. This is a topic of further research.

6.5 Infinite Profile Set Model

Finally, in Table 5 we compare the interaction length, the advice set size, and the average utility shortfall obtained in our experiments with those measured in our previous work, where an infinite number of profiles was considered [3]. In this comparison we use 25000 profiles and we confine the system to use only the add_1 and $trade_{1,2}$ operators to generate new candidate queries: because in [3] the results were obtained by considering only these two editing operations.

We can observe that the interaction length in the two systems is more or less equivalent. The utility shortfall in the proposed finite profiles model is always a bit larger than in the infinite model. This is what has to be paid for the constraining assumption that the number of possible user profiles is finite. The major beneficial effect of the proposed approach is the significant reduction of the advice set size by more than 10 times. Moreover, computing the advice in our implementation, took just some miliseconds, even if 25000 profiles were used, while with the infinite model it required on average some seconds.

In conclusion, we believe that in real scenarios approximating the set of all possible user's utility functions with a finite set is a reasonable assumption, and the small cost paid in terms of increased utility shortfall is compensated by the strong reduction in the size of the advice set and computational complexity, making it feasible for the user to browse the advice set and pick her best query.

Table 5: Comparison of the system performance between the current finite set of profiles model and the infinite model.

DB	Averaged measures	Infinite model	Finite model
Cork	Interaction length	6.09	5.63
	Advice set size	69.88	4.81
	Utility shortfall	0	0.003
Marriott	Interaction length	4.67	3.98
	Advice set size	45.96	5.08
	Utility shortfall	0	0.001
Trentino	Interaction length	5.55	6.31
	Advice set size	59.02	5.17
	Utility shortfall	0	0.037

7. RELATED WORK

Recommending personalized query revisions was first proposed in [3] and then extended in [9]. This approach has proved to be effective, and provides good query recommendations and final product recommendations. It guides the user to the query that selects the products with maximal utility in a short number of query revision interactions. The cited papers describe the details of this approach: the query language, the user model, and the inferences made by the system, observing the user's query revisions and finally the computation of the query suggestions for the user. [3, 9] left open some questions mostly related to the efficiency of computing the query suggestions and the size of the advice set. That approach uses linear programming extensively and require too much computation time to be exploited in an online application. Moreover, the average number of queries suggested to the user at each interaction step is in many cases too large to be presented to a user.

These problems were initially tackled in a preliminary workshop paper [2] by assuming that the user utility function is not an arbitrary one (i.e., coming from an infinite set) but is drawn from a finite set of user profiles that are known by the system. This set represents the possible different users that the system considers that it may interact with. This assumption simplifies the computation of query suggestions (as was also shown here). Moreover, the average number of query suggestions made at each interaction step is also dramatically reduced (by a factor of 10). However, it remained the case that, during the initial steps of query suggestion (when the system knowledge about the user preferences is poor), the number of queries suggested can still be high. Moreover, the authors artificially assumed that the true user utility function is included among the finite set of user profiles contemplated by the system. This is a crude simplification since a totally unknown user approaching the system may have an arbitrary profile and the system has no knowledge about that. We have lifted that assumption in this paper and we have also extended the type of query editing operations, showing that this set can be arbitrarily defined by the system designer.

Critiquing is a conversational recommendation approach that is related to our technique [6]. In critiquing the user is offered query revisions in the form of critiques to the current selected product. The main difference with our proposed approach to building conversational recommender systems is that the query processing in critiquing is based on similarity-based retrieval, while here we are using a logic based approach. Interestingly, in [12, 8] the authors use a multi-attribute utility-based preference model and critiquing suggestion technique that has similar objectives to our approach. They maintain for each user an estimated profile (utility function). Then, they generate the best critiques using the estimated user utility and update the estimated profile by increasing the importance of a feature (weight) if the selected product has a larger feature value compared to

the previously selected one.

Similarly to the "dominated query" concept considered in our work, in [11, 10] the authors consider a conversational recommender system based on example-critiquing that suggest the top K options with highest likelihood to be "not dominated" by others options (Pareto optimality) [4]. The suggestions are based on an analysis of the user's current preference model (adapted in each interaction) and the user's reaction to the suggestions. In our case we take into account the query submitted by the user (user's reaction) in order to generate new queries, and only those that prove to have the highest utility according to the user model inferred so far are considered as not dominated, and thus suggested to the user.

Reducing the number of user-system interaction in finding the target products has been approached in critiquing-based systems through the use of compound critiques [7, 12] which enable the user to express her preferences on multiple attributes at the same time, potentially shortening the interaction cycles. In our approach we enable the user to express implicitly her preferences requesting more than one feature at a time, which reduces the number of cycles needed to reach the best product for the user and making inferences on the true user model is kept simple.

8. CONCLUSIONS AND FUTURE WORK

In this paper we have described and analyzed the performance of a new type of conversational recommender system that suggests query revisions to help the user to find products with the largest utility. We assume that the system contemplates only a finite set of possible user profiles, and interacts with a user who has an unknown profile (probably) close to one of those that the system contemplates.

The results of our experiments show that the finite user profiles set assumption has a strong effect on the process of computing the best query suggestions that guide the user to the products that maximize her utility. In particular the number of user-advisor interaction steps (number of queries issued by the user), and the utility shortfall are low. We have observed a significant reduction in the number of pieces of advice (suggested next queries) provided at each user-advisor interaction step. We have also shown that having a relatively large number of predefined user profiles, and exploiting even simple techniques for filtering the suggestions, is an important ingredient for improving the system performance and producing effective support.

In the current model we consider only Boolean features. But, the proposed approach can be extended to ordinal and numerical features (e.g. hotel category and room price). We plan to develop such an extension in the future. It is also important to note that the user's utility function is assumed to be linear. We plan to investigate the use of more general integral aggregation functions such as Choquet and Sugeno, or Ordering Weighted Averaging functions [1]. This will also be useful for modeling interactions between product features (redundancies, complementarities, contradictions).

Moreover, in this work we have assumed that the user preferences do not change during the interaction with the system and the user is perfectly rational (always selects the best option). In fact, the user may change her preferences or not select the best available option (given her current utility function), and this may generate an inconsistent set of inferred constraints that the system cannot use to produce new query suggestions. We are planning to tackle these issues using relaxation techniques for over-constrained problems [5]. Finally, we must observe that to fully evaluate the proposed approach we must perform live user experiments. Therefore, we are implementing a mobile application for hotel recommendation that exploits the proposed technique.

9. REFERENCES

[1] G. Beliakov, T. Calvo, and S. James. Aggregation of preferences in recommender systems. In *Recommender Systems Handbook*, pages 705–734. 2011.

[2] H. Blanco, F. Ricci, and D. Bridge. Conversational query revision with a finite user profiles model. In *Procs. of the 3rd Italian Information Retrieval Workshop*. CEUR-WS, 2012.

[3] D. Bridge and F. Ricci. Supporting product selection with query editing recommendations. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 65–72, New York, NY, USA, 2007. ACM Press.

[4] B. Faltings and P. Pu. The lookahead principle for preference elicitation: Experimental results. In *In Seventh International Conference on Flexible Query Answering Systems (FQAS*, pages 378–389, 2006.

[5] U. Junker. Quickxplain: Preferred explanations and relaxations for over-constrained problems. In *Proceedings of the 19th National Conference on Artificial Intelligence*, pages 167–172. AAAI Press / The MIT Press, 2004.

[6] L. McGinty and J. Reilly. On the evolution of critiquing recommenders. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 419–453. Springer Verlag, 2011.

[7] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Dynamic critiquing. In *Advances in Case-Based Reasoning, 7th European Conference, ECCBR 2004, Madrid, Spain, August 30 - September 2, 2004, Proceedings*, pages 763–777, 2004.

[8] J. Reilly, J. Zhang, L. McGinty, P. Pu, and B. Smyth. Evaluating compound critiquing recommenders: a real-user study. In *EC '07: Proceedings of the 8th ACM conference on Electronic commerce*, pages 114–123, New York, NY, USA, 2007. ACM.

[9] W. Trabelsi, N. Wilson, D. Bridge, and F. Ricci. Comparing approaches to preference dominance for conversational recommender systems. In E. Gregoire, editor, *Procs. of the 22nd International Conference on Tools with Artificial Intelligence*, pages 113–118, 2010.

[10] P. Viappiani, B. Faltings, and P. Pu. Preference-based search using example-critiquing with suggestions. *J. Artif. Intell. Res. (JAIR)*, 27:465–503, 2006.

[11] P. Viappiani, P. Pu, and B. Faltings. Conversational recommenders with adaptive suggestions. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 89–96. ACM, 2007.

[12] J. Zhang and P. Pu. A comparative study of compound critique generation in conversational recommender systems. In *Procs. of 4th Intl. Conf. on Adaptive Hypermedia & Adaptive Web-Based Systems*, pages 234–243. Springer-Verlag, 2006.

Eliciting Stakeholder Preferences for Requirements Prioritization

Alexander Felfernig
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b
8010 Graz, Austria
afelfern@ist.tugraz.at

Gerald Ninaus
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b
8010 Graz, Austria
gninaus@ist.tugraz.at

Florian Reinfrank
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b
8010 Graz, Austria
freinfra@ist.tugraz.at

ABSTRACT

Requirements engineering is a very critical phase in software development process. Requirements can be interpreted as basic decision alternatives which have to be negotiated by stakeholders. In this paper we present the results of an empirical study which focused on the analysis of key influence factors of successful requirements prioritization. This study has been conducted within the scope of software development projects at our university where development teams interacted with a requirements prioritization environment. The major result of our study is that anonymized preference elicitation can help to significantly improve the quality of requirements prioritization, for example, in terms of the degree of team consensus, prioritization diversity, and quality of the resulting software components.

Categories and Subject Descriptors

D.2 [**Software Engineering**]: Requirements Engineering; D.2.1 [**Requirements/Specifications**]: Requirements Negotiation; H.5 [**Information Interfaces and Presentation**]: Modeling Environments

General Terms

Human Factors, Experimentation

Keywords

Requirements Prioritization, Group Decision Making

1. INTRODUCTION

Requirements Engineering (RE) is the branch of software engineering concerned with the real-world goals for functions of and constraints on software systems [14]. RE is considered as one of the most critical phases in software projects, and poorly implemented RE is one major risk for project failure [8]. Requirements are the basis for all subsequent phases in the development process and high quality requirements are a major precondition for the success of the project [4].

Today's software projects still have a high probability to be canceled or at least to significantly exceed the available resources [13]. As stated by Firesmith [5], the phase of requirements engineering receives rarely more that 2-4% of the overall project efforts although more efforts in getting the requirements right result in significantly higher project success rates. A recent Gartner report [7] states that *requirements defects are the third source of product defects (following coding and design), but are the first source of delivered defects. The cost of fixing defects ranges from a low of approximately $70 (cost to fix a defect at the requirements phase) to a high of $14.000 (cost to fix a defect in production). Improving the requirements gathering process can reduce the overall cost of software and dramatically improve time to market.*

Requirements can be regarded as a representation of decision alternatives or commitments that concern the functionalities and qualities of the software or service [1]. Requirements engineering (RE) is then a complex task where stakeholders have to deal with various decisions [11]:

- *Quality decisions*, e.g., is the requirement non-redundant, concrete, and understandable?
- *Preference decisions*, e.g., which requirements should be considered for the next release?
- *Classification decisions*, e.g., to which topic does this requirement belong?
- *Property decisions*, e.g., is the effort estimation for this requirement realistic?

Stakeholders are often faced with a situation where the amount and complexity of requirements outstrips their capability to survey them and reach a decision [3]. The amount of knowledge and number of stakeholders involved in RE processes tend to increase as well. This makes individual as well as group decisions much more difficult.

The focus of this paper will be *preference decisions*, i.e., we want to support groups of stakeholders in the context of *prioritizing software requirements* for the next release. Typically, resource limitations in software projects are triggering the demand of a prioritization of the defined requirements [8]. Prioritizations support

RecSys'12 9th - 13th September, 2012, Dublin, Ireland
Paper presented at the 2012 Decisions@RecSys workshop in conjunction with the 6th ACM conference on Recommender Systems. Copyright ©2012 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors..

software project managers in the systematic definition of software releases and help to resolve existing preference conflicts among stakeholders.

Only a systematic prioritization can guarantee that the most essential functionalities of the software system are implemented in-time [12]. Typically, requirements prioritization is a collaborative task where stakeholders in a software project collaborate with the goal to achieve consensus regarding the prioritization of a given set of requirements. The earlier requirements are prioritized, the lower is the effort of implementing irrelevant requirements and the higher is the amount of available resources to implement the most relevant requirements [12].

Establishing consensus between stakeholders regarding the prioritization of a given set of requirements is challenging. Prioritizations do not only have to take into account business process related criteria but as well criteria which are related to technical aspects of the software. Especially in larger projects, stakeholders need a tool-supported prioritization approach which can help to reduce influences related to psychological and political factors [12]. Requirements prioritization is a specific type of group work which becomes increasingly important in organizations [10].

Prioritization decisions are typically taken in groups but this task is still ineffective due to reasons such as social blocking, censorship, and hidden agendas [10]. *One balancing strategy is to drop or defer low priority requirements to a later release* [12]. In a study conducted at the Graz University of Technology during the course *Object Oriented Analyses and Design*, the stakeholder part of the customer was impersonated by four course assistants. These assistants were not aware of the study settings and had to review the software functionality developed by the different teams. This evaluation did not include a code review. Rather it was supposed to assess the user experience of the product and which important functionality was supported. These important functions were partially defined by the exercise given in the course. The result of this evaluation is represented by a quality value between 0 and 30 credits. The major contribution of this paper is to show how *anonymity* in group decision processes can help to improve the quality of requirements prioritizations.

The remainder of this paper is organized as follows. In Section 2 we provide an overview of the basic functionalities of the IntelliReq requirements engineering environment developed at our university to collect preferences and decisions of stakeholders during the course *Object-oriented Analysis and Design* at the Graz University of Technology. In Section 3 we introduce the basic hypotheses that have been investigated within the scope of our empirical study; in this context we also provide details about the study design. In Section 4 we report the major results of our empirical study and show the effect of anonymity on the group consensus, the decision diversity and the output quality. With Section 5 we conclude the paper.

2. INTELLIREQ DECISION SUPPORT

IntelliReq is a group decision environment that supports computer science students at our university in deciding on which requirements should be implemented within the scope of their software projects. For this task 219 students enrolled in a course about *Object-Oriented Analyse and Design* at the Graz University of Technology had to form groups of 5–6 members. Unfortunately, it is not possible to evaluate the existing knowledge and experience of

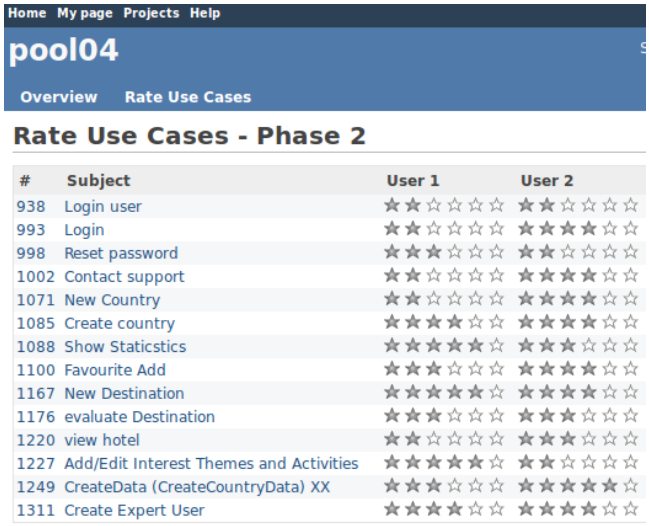


Figure 1: IntelliReq Anonymous Preference Presentation: the preferences of users are anonymized by replacing the stakeholder names with the terms "User 1", "User 2", ... , "User n". The order of stakeholders and the assignment to these terms is randomly generated.

the students and the resulting groups. We therefore distributed the resulting groups randomly on the different evaluation pools and assume that the knowledge and experience is equally distributed on each pool. Each group had to implement a software system with an average effort of about 8 man months. Figure 1 shows the anonymized preference presentation of stakeholders in **IntelliReq**.

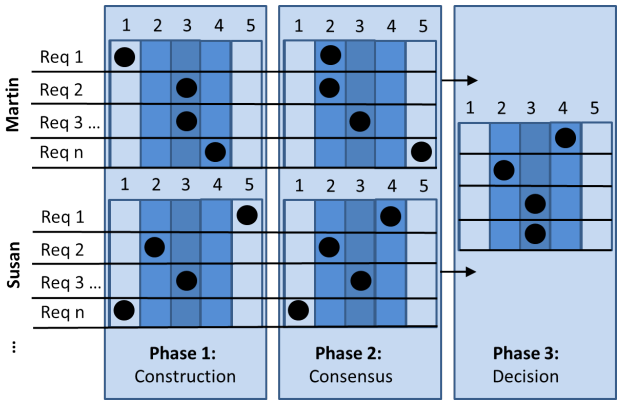


Figure 2: IntelliReq Prioritization (Decision) Process. Construction: stakeholders define their initial preferences; Consensus: stakeholders adapt their preferences on the basis of the knowledge about preferences of other stakeholders. Decision: project managers take the final group decision.

In our study, 39 software development teams had to define a set of requirements which in the following had to be implemented. These requirements had to be prioritized and the resulting prioritization served as a major criteria for evaluating the corresponding software components at the end of the project.

The requirements prioritization process consisted of three different

phases (see Figure 2) denoted as *construction* (collection of individual stakeholder preferences), *consensus* (adaptation of own preferences, see Figure 3), and *decision* (group decision defined and explained by the project manager). This decision process structure results in about 15.000 stakeholder decisions and 798 corresponding group decisions taken by the team leaders (project managers). On the basis of this scenario we conducted an empirical evaluation with the goal to analyze the effects of supporting anonymized requirements prioritization. The basic settings of this study will be presented in the following section.

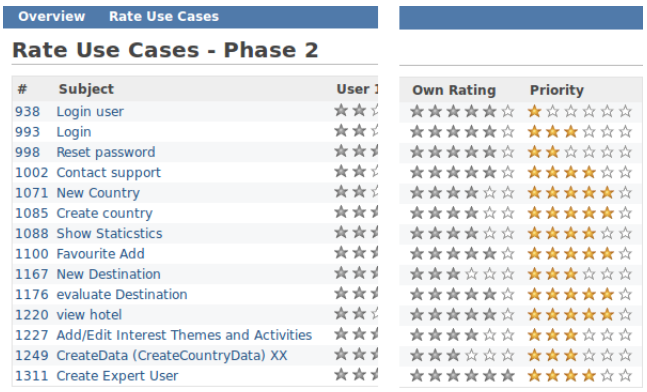


Figure 3: INTELLIREQ Preference Adjustment (Consensus): stakeholders can view their initial preferences and preferences of other stakeholder. With this information stakeholders can adjust their preferences to increase group consensus.

3. EMPIRICAL STUDY

Within the scope of our empirical study we wanted to investigate the impact of *anonymous preference elicitation* on the decision support quality of the INTELLIREQ environment. Consequently, each project team interacted with exactly one of two existing types of preference elicitation interface. One interface (*type 1: non-anonymous preference elicitation*) provided an overview of the personal preferences of team members (stakeholders) where each team member was represented by her/his name. In the second type of interface (*type 2: anonymous preference elicitation*) the preferences of team members were shown in anonymized form where the name of the individual team member was substituted with the terms "User 1", "User 2", etc (see Figure 1). The hypotheses (H1–H8) used to evaluate the decision process are summarized in Figure 4. These hypotheses were evaluated on the basis of the following observation variables.

Anonymous preference elicitation. This variable indicates with which type of prioritization interface the team members were confronted (either summarization of the preferences of the team members including the name of the team members or not including the name of the team members).

Consensus and Dissent. An indication to which extent the team members managed to achieve consensus (dissent) – see the second phase of the group decision process in Figure 2 – is provided by the corresponding variables. We measured the *consensus of a group* on the basis of the standard deviation derived from requirement-specific group decisions. Formula 1 can be used to determine the *dissent* of a group x which is defined in terms of the normalized sum of the standard deviations (sd) of the requirement-specific votings. The group *consensus* can then be interpreted as the counterpart of dissent (see Formula 2). As the consensus is the simple inversion of the dissent, we will only take into account the consensus in the remaining paper.

$$dissent(x) = \frac{\sum_{r \in Requirements} sd(r)}{|Requirements|} \quad (1)$$

$$consensus(x) = \frac{1}{dissent(x)} \quad (2)$$

Decision Diversity. The decision diversity of a group can be defined in terms of the average over the decision diversity of individual users in the consensus phase (see Figure 2). The latter is defined in terms of the standard deviation derived from the decision d_u of a user – a decision consists of the individual requirements prioritizations of the user.

$$diversity(x) = \frac{\sum_{u \in Users} sd(d_u)}{|Users|} \quad (3)$$

Output Quality. The output quality of the software projects conducted within the scope of our empirical study has been derived from the criteria such as degree of fulfillment of the specified requirements. We also weighted the requirements according to their defined priority in the prioritization task. E.g. not including a very high important requirement enormously decreases the quality value. On the opposite, low priority requirements will only have a small impact on the quality value. Therefore, defining a high priority for a requirement which is of minor importance has to be implemented anyway. On the other hand, each group has to implement all important requirements for the user experience and which are important for the functionality. Therefore, the requirements prioritization has a direct impact on the quality value. The quality of the project output has been determined by teaching assistants who did not know to which type of preference elicitation interface (anonymous vs. non-anonymous) the group has been assigned to. These assignments were randomized over all teaching assistants, i.e., each teaching assistant had to evaluate (on a scale of 0..30 credits) groups who interacted with an anonymous and a non-anonymous interface.

Within the scope of our study we wanted to evaluate the following hypotheses.

H1: Anonymous Preference Elicitation increases Consensus. The idea behind this hypothesis is that anonymous preference elicitation helps to decrease the commitment [2] related to an individual decision taken in the preference construction phase (see Figure 2), i.e., changing his/her mind is easier with an anonymous preference elicitation interface. Furthermore, anonymous preference elicitation increases the probability of detecting hidden profiles [6], i.e., increases the probability of exchanging decision-relevant information [9].

H2: Anonymous Preference Elicitation decreases Dissent. Following the idea of hypothesis H1, non-anonymous preference elicitation increases commitment with regard to already taken (and published) decisions. It also decreases the probability of detecting hid-

den profiles [6] and thus also decreases the probability of high-quality decisions (see H3).

H3: Consensus increases Decision Diversity. As a direct consequence of an increased exchange of decision-relevant information (see Hypothesis H1), deep insights into major properties of the decision problem can be expected. As a consequence, the important differentiation between important, less important, and unimportant requirements with respect to the next release [3] can be achieved.

H4: Dissent decreases Decision Diversity. From less exchange of decision-relevant information we can expect a lower amount of globally available decision-relevant information. As a consequence, the differentiation between important, less important, and unimportant requirements is a bigger challenge for the engaged stakeholders.

H5: Consensus increases Output Quality. From Hypothesis H3 we assume a positive correlation between the degree of consensus and the diversity of the group decision. The diversity is an indicator for a meaningful triage [3] between important, less important, and unimportant requirements.

H6: Dissent decreases Output Quality. In contrary, dissent leads to a lower decision diversity and – as a consequence – to less meaningful results of requirements triage.

H7: Decision Diversity increases Output Quality. Group decision diversity is assumed to be a direct indicator for the quality of the group decision. With this hypothesis we want to analyze the direct interrelationship between prioritization diversity and the quality of the resulting software.

H8: Anonymous Preference Elicitation increases Output Quality. Finally, we want to explicitly analyze whether there exists a relationship between the type of preference elicitation and the corresponding output quality.

4. STUDY RESULTS

We analyzed the hypotheses (H1–H8) on the basis of the variables introduced in Section 3.[1] We used a Mann-Whitney U-test if the examined data set is not normal distributed (H1,H2) and the t-test if the data set is normal distributed (H8). The correlations (H3 – H7) are calculated with Pearson correlations (normal distribution) and with the Spearman's rank correlations (no normal distribution).

H1. The degree of group consensus in teams with anonymous preference elicitation is significantly higher compared to teams with non-anonymous preference elicitation (Mann-Whitney U-test, $p < 0.05$). An explanation model can be the reduction of commitment [2] and a higher probability of discovering hidden profile information which improves the overall knowledge level of the team.

H2. Group dissent is an inverse function of group consensus and – as a consequence – teams with non-anonymous preference elicitation have a significantly higher dissent (Mann-Whitney U-test, $p < 0.05$). In this context, non-anonymous preference elicitation can lead to higher commitment with regard to the orginially articulated preferences.

[1]We are aware of the fact that *dissent* is the inverse function of *consensus*, however, for reasons of understandability we decided to explicitly include *dissent* as a decision variable.

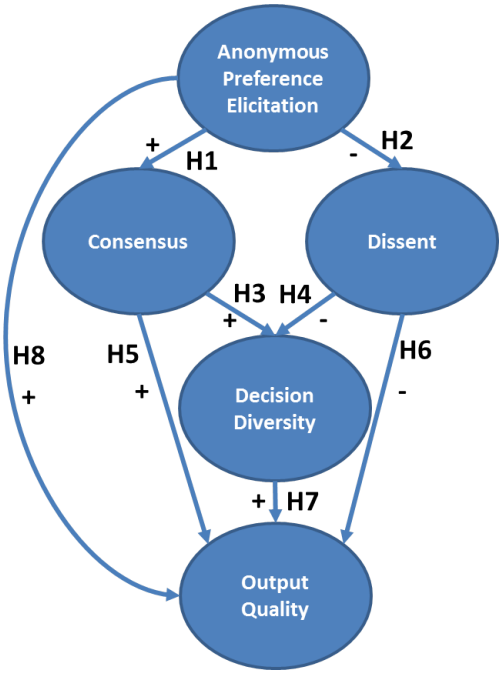


Figure 4: Hypotheses defined to evaluate the INTELLIREQ Decision Support.

H3. There is a positive correlation between the group consensus and the corresponding decision diversity (correlation 0.523, $p < 0.01$). More group discussions can lead to a higher level of relevant knowledge about the decision problem. In the following this can lead to a development of a deeper understanding of the need of requirements triage [3] which leads to a higher degree of decision diversity.

H4. Dissent is an inverse function of group consensus – the higher the dissent, the lower the corresponding decision diversity (correlation -0.523, $p < 0.01$). A lower degree of group decision diversity (prioritization diversity) can be explained by a lower degree of decision-relevant knowledge.

H5. Consensus in group decision making increases the output quality (correlation 0.399, $p < 0.01$). An overlap in the personal stakeholder preferences can be interpreted as an indicator of a common understanding of the underlying set of requirements. This leads to a better prioritization and a higher quality of the resulting software components.

H6. The hypothesis can be confirmed (correlation -0.399, $p < 0.01$), i.e., there is a negative correlation between group dissent and the corresponding output quality.

H7. In our analysis we could detect a positive correlation between group decision diversity (diversity of prioritization) and the corresponding output quality (correlation 0.311, $p < 0.01$). Decision diversity can be seen as an indicator of a reasonable triage process and reasonable prioritizations result in higher-quality software components.

H8. Groups with anonymous preference elicitation performed sig-

nificantly better compared to groups with a non-anonymous preference elicitation (independent two-sample t-test, $p < 0.05$).

5. CONCLUSIONS

Requirements prioritization is an important task in software development processes. In this paper we motivated the application of requirements prioritization and discussed issues related to the aspect of anonymizing group decision processes in requirements prioritization. The results of our empirical study clearly show the advantages of applying anonymized preference elicitation, for example in terms of higher-quality software components, and can be seen as a step towards a more in-depth integration of decision-oriented research in requirements engineering processes.

6. REFERENCES

[1] A. Aurum and C. Wohlin. The fundamental nature of requirements engineering activities as a decision-making process. *Information and Software Technology*, 45(14):945–954, 2003.

[2] R. Cialdini. The science of persuasion. *Scientific American*, 284:76–81, 2001.

[3] A. Davis. The art of requirements triage. *IEEE Computer*, 36(3):42–49, 2003.

[4] A. Felfernig, C. Zehentner, G. Ninaus, H. Grabner, W. Maaleij, D. Pagano, L. Weninger, and F. Reinfrank. Group decision support for requirements negotiation. In *Advances in User Modeling*, pages 105–116, 2012.

[5] D. Firesmith. Prioritizing requirements. *Journal of Object Technology*, 3(8):35–47, 2004.

[6] T. Greitemeyer and S. Schulz-Hardt. 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 & Social Psychology*, 84(2):332–339, 2003.

[7] G. Group. Hype cycle for application development: Requirements elicitation and simulation. 2011.

[8] H. Hofmann and F. Lehner. Requirements engineering as a success factor in software projects. *IEEE Software*, 18(4):58–66, 2001.

[9] A. Mojzisch and S. Schulz-Hardt. Knowing other's preferences degrades the quality of group decisions. *Journal of Personality & Social Psychology*, 98(5):794–808, 2010.

[10] A. Pinsonneault and N. Heppel. Anonymity in group support systems research: A new conceptualization, measure, and contingency framework. *Journal of Management Information Systems*, 14:89–108, 1997.

[11] B. Regnell, B. Paech, C. Aurum, C. Wohlin, A. Dutoit, and J. ochDag. Requirements means decision! In *1st Swedish Conf. on Software Engineering and Practice (SERP'01)*, pages 49–52, Innsbruck, Austria, 2001.

[12] K. Wiegers. First things first: Prioritizing requirements. *Software Development*, 1999.

[13] D. Yang, D. Wu, S. Koolmanojwong, A. Brown, and B. Boehm. Wikiwinwin: A wiki based system for collaborative requirements negotiation. In *HICCS 2008*, page 24, Waikoloa, Big Island, Hawaii, 2008.

[14] P. Zave. Classification of research efforts in requirements engineering. *ACM Computing Surveys*, 29(4):315–321, 1997.

Recommendation systems in the scope of opinion formation: a model

Marcel Blattner
Laboratory for Web Science
University of Applied Sciences FFHS
Regensdorf, Switzerland
marcel.blattner@ffhs.ch

Matus Medo
Physics Department
University of Fribourg
Fribourg, Switzerland
matus.medo@unifr.ch

ABSTRACT

Aggregated data in real world recommender applications often feature fat-tailed distributions of the number of times individual items have been rated or favored. We propose a model to simulate such data. The model is mainly based on social interactions and opinion formation taking place on a complex network with a given topology. A threshold mechanism is used to govern the decision making process that determines whether a user is or is not interested in an item. We demonstrate the validity of the model by fitting attendance distributions from different real data sets. The model is mathematically analyzed by investigating its master equation. Our approach provides an attempt to understand recommender system's data as a social process. The model can serve as a starting point to generate artificial data sets useful for testing and evaluating recommender systems.

Categories and Subject Descriptors

H.1.m [**Information Systems**]: Miscellaneous

General Terms

Experimentation, Theory

Keywords

recommender systems, opinion formation, complex networks

1. INTRODUCTION

This is the information age. We are witnessing information production and consumption in a speed never seen before. The WEB2.0 paradigm enables consumers and producers to exchange data in a collaborative way benefiting both parties. However, one of the key challenges in our digitally-driven society is information overload [7]. We have the 'pain of choice'. Recommendation systems represent a possible solution to this problem. They have emerged as a research area on its own in the 90s [42, 20, 28, 21, 11]. The interest in recommendation systems increased steadily in recent years, and attracted researchers from different fields [43]. The success of highly rated Internet sites as Amazon, Netflix, YouTube, Yahoo, Last.fm and others is to a large extent based on their recommender engines. Corresponding applications recommend everything from CD/DVD's, movies, jokes, books, web sites to more complex items such as financial services.

The most popular techniques related to recommendation systems are collaborative filtering [8, 26, 11, 24, 28, 21, 41, 45] and content-based filtering [14, 40, 35, 5, 30]. In addition, researchers developed alternative methods inspired by fields as diverse as machine learning, graph theory, and physics [16, 17, 37, 52, 51, 10, 48, 50]. Furthermore, recommendation systems have been investigated in connection with trust [2, 39, 47, 32, 33] and personalized web search [9, 12, 46], which constitutes the new research frontier in search engines.

However, there are still many open challenges in the research field of recommendation systems [1, 22, 25, 18, 24, 43, 15]. One key question is connected to the understanding of the user rating mechanism. We build on a well documented influence of social interactions with peers on the decision to vote, favor, or even purchase an item [44, 27]. We propose a model inspired by opinion formation taking place on a complex network with a predefined topology. Our model is able to generate data observed in real world recommender systems. Despite its simplicity, the model is flexible enough to generate a wide range of different patterns. We mathematically analyze the model using a mean field approach to the full Master Equation. Our approach provides an understanding of the data in recommender systems as a product of social processes. The model can serve as a data generator which is valuable for testing and evaluation purposes for recommender systems.

The rest of the paper is organized as follows. The model is outlined in Sec. (2). Methods, data set descriptions, and validation procedures are in Sec. (3). Results are presented in Sec. (4). Discussion and an outlook for future research directions are in Sec. (5).

2. MODEL

2.1 Motivation

Our daily decisions are heavily influenced by various information channels: advertisement, broadcastings, social interactions, and many others. Social ties (word-of-mouth) play a pivotal role in consumers buying decisions [44, 27]. It was

Paper presented at the 2012 Decisions@RecSys workshop in conjunction with the 6th ACM conference on Recommender Systems. Copyright 2012 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

demonstrated by many researchers that personal communication and informal information exchange not only influence purchase decisions and opinions, but shape our expectations of a product or service [49, 4, 3]. On the other hand, it was shown [23], that social benefits are a major motivation to participate on opinion platforms. If somebody is influenced by recommendations on an opinion platform like MovieLens or Amazon, social interactions and word-of-mouth in general are additional forces governing the decision making process to purchase or even to rate an object in a particular way [31].

Our model is formulated within an opinion formation framework where social ties play a major role. We shall discuss the following main ingredients of our model:

- Influence-Network (IN)
- Intrinsic-Item-Anticipation (IIA)
- Influence-Dynamics (ID)

Influence Network.

We call the network where context-relevant information exchange takes place an Influence-Network (IN). Nodes of the IN are people and connections between nodes indicate the influence among them. Note that we put no constraints on the nature of how these connections are realized. They may be purely virtual (over the Internet) or based on physical meetings. We emphasize that INs are domain dependent, i.e., for a given community of users, the Influence Network concerning books may differ greatly (in topology, number of ties, tie strength, etc.) from that concerning another subject such as food or movies. Indeed, one person's opinion leaders (relevant peers) concerning books may be very different from those for food or other subjects. In this scope, we see the INs as domain-restricted views on social networks. It is thus reasonable to assume that Influence Networks are similar to social interaction networks which often exhibit a scale-free topology [6]. However, our model is not restricted to a particular network structure.

Intrinsic-Item-Anticipation.

Suppose a new product is launched on the market. Advertisement, marketing campaigns, and other efforts to attract customers predate the launching process and continue after the product started to spread on the market. These efforts influence product-dependent customer anticipation. It is clear that the resulting anticipation is a complex combination of many different components including intrinsic product quality and possibly also suggestions from recommendation systems.

In our model we call the above-described anticipation Intrinsic-Item-Anticipation (IIA) and measure it by a single number. It is based on many external sources, except for the influence generated by social interactions. It is the opinion on something taken by individuals, before they start to discuss the subject with their peers. Furthermore, we assume that an individual will invest resources (time/money) into an object only, if the Intrinsic-Item-Anticipation is above a particular threshold, which we call Critical-Anticipation-Threshold.

Influence-Dynamics.

The Influence-Dynamics describes how individuals' Intrinsic-Item-Anticipations are altered by information exchange via the connections of the corresponding Influence-Network. From our model's point of view this means the following: an individual's IIA for a particular item i may be shifted due to social interactions with directly connected peers (these interactions thus take place on the corresponding IN), who already experienced the product or service in question. This process can shift the Intrinsic-Item-Anticipation of an individual who did not yet experience product/object i closer to or beyond the critical-anticipation-threshold.

We now summarize the basic ingredients of our model. An individual user's opinions on objects are assembled in two consecutive stages: i) opinion making based on different external sources, including suggestions by recommendation systems and ii) opinion making based on social interactions in the Influence-Network. The second process may shift the opinions generated by the first process.

2.2 Mathematical formulation of the model

In this section we firstly describe how individuals' Intrinsic-Item-Anticipations may change due to social interactions taking place on a particular Influence-Network. Secondly, we introduce dynamical processes governing the opinion propagation.

IIA shift.

We model a possible shift in the IIA as:

$$\hat{f}_{ij} = f_{ij} + \left[\frac{\Theta_j}{k_j}\right]^{(1-\gamma)}. \tag{1}$$

where $\hat{f}_{ij}$ is the shifted Intrinsic-Item-Anticipation of individual j for object i, f_{ij} is the unbiased IIA, Θ_j is the number of j's neighbors, who already experienced and liked item i, k_j denotes the total number of j's neighbors in the corresponding IN, and $\gamma \in (0, 1)$ quantifies trust of individuals to their peers. An individual j will consume, purchase, or positively rate an item i only if

$$\hat{f}_{ij} \geq \Delta. \tag{2}$$

We identify Δ as the Critical-Anticipation-Threshold. Values of f_{ij} are drawn from a probability distribution f_i. Since the IIA for each individual is an aggregate of many different and largely independent contributions, we assume that f_i is normally distributed, $f_i \in \mathcal{N}(\mu_i, \sigma)$. (Unless stated otherwise.) To mimic different item anticipations for different objects i, we draw the mean μ_i from a uniform distribution $U(-\epsilon, \epsilon)$. We maintain μ_i, ϵ, and σ, so that f_i is roughly bounded by $(-1, 1)$, i.e., $-1 \leq \mu - 3\sigma < \mu + 3\sigma \leq 1$. Note that $\hat{f}_{ij}$ can exceed these boundaries after a shift of the corresponding IIA occurs. The second term on the right hand side of Eq. (1) is the influence of j's neighborhood weighted by trust γ. To better understand the interplay between γ and the density of attending users in the neighborhood of user i, $\rho := \Theta_j/k_j$, we refer to Fig. 1. Trust $\gamma \approx 1$ causes a big shift on the IIA's even for $\rho \approx 0$. On the other hand, $\gamma \approx 0$ needs high ρ to yield a significant IAA shift. These properties are understood as follows: people trusting strongly in their peers need only few positive opinions to be convinced, whereas people trusting less in their social environment need considerable more signals to be influenced.

Influence-Dynamics.

The Influence-Dynamics proceeds as follows. Firstly, we

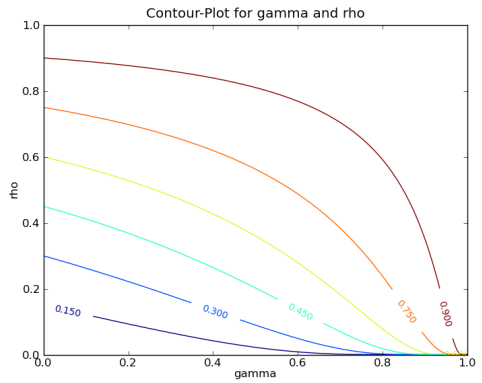


Figure 1: Contour plot for γ and $\rho = \Theta_j/k_j$. Numbers inside the plot quantify the shift in the IAA as a function of γ and ρ.

draw an Influence-Network IN($\mathcal{P}$) with a fixed network topology (power-law, Erdős-Rényi, or another). $\mathcal{P}$ refers to a set of appropriate parameters for the Influence-Network in question (like network type, number of nodes, etc.). The network's topology is not affected by the dynamical processes (opinion propagation) taking place on it. We justify this static scenario by assuming that the time scale of the topology change is much longer then the time scale [1] of opinion spreading in the network. Each node in the Influence-Network corresponds to an individual. For each individual j we draw an unbiased Intrinsic-Item-Anticipation f_{ij} from the predefined probability distribution f_i. At each time step, every individual is in one of the following states: $\{S, A, D\}$. S refers to a susceptible state and corresponds to the initial state for all nodes at $t = 0$. A refers to an attender state and corresponds to an individual with the property $\hat{f}_{ij} \geq \Delta$. D refers to a denier state with the property $\hat{f}_{ij} < \Delta$ after an information exchange with his/her peers in the Influence-Network happened. An individual in state D or A can not change his/her state anymore. It is clear that an individual in state A cannot back transform to the susceptible state S, since he/she did consume or favor item i and we do not account for multiple attendances in our model. An individual in state D was influenced but not convinced by his opinion leaders (directed connected peers). We make the following assumption here: if individual j's opinion leaders are not able to convince individual j, meaning that individual's j Intrinsic Item Anticipation $\hat{f}_{ij}$ stays below the critical threshold Δ after the influence process, then we assume that j's opinion not to attend object i remains unchanged in the future. Therefore we have the following possible transitions for each node in the influence network: $j_S \to j_A$ or $j_S \to j_D$. Node states are updated asynchronously which is more realistic than synchronous updating, especially in social interaction models [13]. The Influence-Dynamics is summarized in Algorithm 1.

Algorithm 1 RecSysMod algorithm. $\mathcal{P}$ contains the configuration parameter for the network. Δ is the Anticipation Threshold and γ denotes the trust. $O \in \mathbb{N}$ is the number of objects to simulate. $G(N, E)$ is the network. N is the set of nodes and E is the set of edges.

procedure RecSysMod_I($\mathcal{P}, \Delta, \gamma, O$)
  $G(N, E) \leftarrow$ GenNetwork($\mathcal{P}$)
  for all Objects in O **do**
    generate distribution f_i from $\mathcal{N}(\mu_i, \sigma)$
    for each node $j \in N$ in G **do**
      draw f_{ij} from f_i
      if $f_{ij} < \Delta$ **then**
        $j_{state} \leftarrow S$
      else
        $j_{state} \leftarrow A$
      end if
    end for
    repeat
      for all j with $j_{state} = S$ AND $\Theta_j > 0$ **do**
        $\hat{f}_{ij} \leftarrow f_{ij} + \left[\frac{\Theta_j}{k_j}\right]^{(1-\gamma)}$
        if $\hat{f}_{ij} < \Delta$ **then**
          $j_{state} \leftarrow D$
        else
          $j_{state} \leftarrow A$
        end if
      end for
    until $|\{j | j_{state} = S \text{ AND } \Theta_j > 0\}| = 0$
  end for
end procedure

Master Equation.

We are now in the position to formulate the Master Equation for the dynamics. As already said before, two things can happen when a non-attender is connected to an attender: a) he/she becomes an attender too, or b) he/she becomes a denier who will not attend/favor the item. For these two interaction types we formally write:

$$\begin{aligned} S + A &\xrightarrow{\lambda} 2A \\ S + A &\xrightarrow{\alpha} D + A \end{aligned} \tag{3}$$

Here λ denotes the probability that a susceptible node connected to an attender becomes an attender too, and α is the probability that a susceptible node attached to an attender becomes a denier. To take into account the underlying network topology of the Influence Network it is common to introduce compartments k [19]. Let N_k^A be the number of nodes in state A with k connections, N_k^S the number of nodes in state S with k connections, and N_k^D the number of nodes in state D with k connections, respectively. Furthermore we define the corresponding densities: $a_k(t) = N_k^A/N_k$, $s_k(t) = N_k^S/N_k$ and $d_k(t) = N_k^D/N_k$. N_k is the total number of nodes with k connections in the network. Since every node from N_k must be in one of the three states, $\forall t: \; a_k(t) + s_k(t) + d_k(t) = 1$. A weighted sum over all k compartments gives the total fraction of attenders at time t, $a(t) = \sum_k P(k) a_k(t)$ where $P(k)$ is the degree distribution of the network (it also holds that $a(t) = N^A(t)/N$). The time dependence of our state variables $a_k(t), d_k(t), s_k(t)$ is

$$\left.\begin{aligned} \dot{a}_k(t) &= \lambda k s_k(t) \Omega \\ \dot{d}_k(t) &= \alpha k s_k(t) \Omega \\ \dot{s}_k(t) &= -(\alpha + \lambda) k s_k(t) \Omega \end{aligned}\right\} \tag{4}$$

[1] The term time scale denotes a dimensionless quantity and specifies the devisions of time. A shorter time scale means a faster spreading of opinions in the network.

where Ω is the density of attenders in the neighborhood of susceptible node with k connections averaged over k

$$\Omega = \sum_k P(k)(k-1)a_k / \langle k \rangle \tag{5}$$

where $\langle k \rangle$ denotes the mean degree of the network. As outlined above, λ is the probability that a node in state S transforms to state A if it is connected to a node in state A. This happens when $\hat{f}_{ij} > \Delta$. Therefore, we have $\Delta_- < f_{ij} < \Delta$ where $\Delta_- = \Delta - (1/k)^{1-\gamma}$. From this we have $\lambda = \int_{\Delta_-}^{\Delta} f(x)dx$, where $f(x)$ is the expectation distribution. Similarly we write for $\alpha = \int_l^{\Delta_-} f(x)dx$, where l denotes the lower bound of the expectation distribution $f(x)$. A crude mean field approximation can be obtained by multiplying the right hand sides of Eq. (4) with $P(k)$ and summing over k, which yields a set of differential equations

$$\left.\begin{aligned} \dot{a}(t) &= \lambda \langle k \rangle s(t)a(t), \\ \dot{d}(t) &= \alpha \langle k \rangle s(t)a(t), \\ \dot{s}(t) &= -(\alpha + \lambda) \langle k \rangle s(t)a(t). \end{aligned}\right\} \tag{6}$$

which is later used to obtain analytical results for the attendance fraction $a(t)$.

3. METHODS

We describe here our simulation procedures, datasets, experiments, and analytical methods.

Simulations.

Our simulations employ Alg. (1). As outlined in the model section, we do not change the network topology during the dynamical processes. We experiment with two different network types, Erdős-Rényi (ER), and power law (PL) which are both generated by a so-called configuration model [34]. ER and PL represent two fundamentally different classes of networks. The former is characterized by a typical degree scale (mean degree of the network), whereas the latter exhibits a fat-tailed degree distribution which is scale free. The networks are random and have no degree correlations and no particular community structure. To obtain representative results we stick to the following approach: we fix the network type, number of nodes, number of objects, and network type relevant parameters to draw an ER or PL network. We call this a configuration $\mathcal{P}$. In addition, we fix the variance σ of the anticipation distributions f_i. We perform each simulation on 50 different networks belonging to the same configuration $\mathcal{P}$ and on each network we simulate the dynamics 50 times. Then we average the obtained attendance distributions over all 2500 simulations.

Datasets.

To show the validity of our model we use real world recommender datasets. **MovieLens** (movielens.umn.edu), a web service from GroupLens (grouplens.org) where ratings are recorded on a five stars scale. The data set contains 1682 movies and 943 users. Only $6,5\%$ of possible votes are expressed. **Netflix** data set (netflix.com). We use the Netflix grand prize data set which contains 480189 users and 17770 movies and also uses a five stars scale. **Lastfm** data set (Lastfm.com). This data set contains social networking, tagging, and music artist listening information from users of the Last.fm online music system. There are 1892 users, 17632 artists, and 92834 user-listended artists relations in total. In addition, the data set contains 12717 bi-directional user friendship relations. These data sets are chosen because they exhibit very different attendance distributions and thus provide an excellent playground to validate our model in different settings.

Experiments.

Data topologies. We firstly investigate the simulated attendance distributions as a function of trust γ, the anticipation threshold Δ, and the network topology. For this purpose we simulate the dynamics on a toy network with 500 nodes and record the final attendance number of 300 objects. The simulation is conducted for ER and PL networks and performed as outlined in the simulations paragraph above. In Fig.(2) and Fig.(3) we investigate the skewness [53] of the attendance distributions and the maximal attendance obtained for the corresponding parameter settings. The skewness of a distribution is a measure for the asymmetry around its mean value. A positive skewness value means that there is more weight to the left from the mean, whereas a negative value indicates more weight in the right from the mean.

Fitting real data. We explore the model's ability to fit real world recommendation attendance distributions found in the described data sets. For this purpose we fix for the Netflix data set a network with 480189 nodes and perform a simulation for 17770 objects. In the MovieLens case we do the same for 943 nodes and 1682 objects and for the Lastfm data set we simulate on a network with 1892 nodes and 17632 objects. In the case of Lastfm we have the social network data as well. We validate our model on that data set by two experiments: a) we use the provided user friendship network as simulation input and fit the attendance distribution and b) we fit the attendance distribution like in the MovieLens and Netflix case with an artificially generated network.

Mathematical analysis. We investigate the Master Equations Eq. (4) and Eq. (6). We provide a full analytical solution for Eq. (6) and an analytical approximation for Eq. (4) in the early spreading stage.

4. RESULTS

Data topologies. The landscape of attendance distributions of our model is demonstrated in Fig. (2) and Fig. (3). To obtain these results, simulations were performed as described in Sec. (3). The item anticipation f_i was drawn from a normal distribution with mean values $\mu_i \in U(-0.1, 0.1)$ and variance $\sigma = 0.25$ fixed for all items. Both networks have 500 nodes. In the Erdős-Rényi case, we used a wiring probability $p = 0.03$ between nodes. The Power Law network was drawn with an exponent $\delta = 2.25$. The simulated attendance distributions in Fig.(2) and Fig.(3) show a wide range of different patterns for both ER and PL Influence-Networks. In particular, both network types can serve as a basis for attendance distributions with both positive and negative skewness. Therefore, the observed fat-tailed distributions are not a result of the heterogeneity of a scale free network but they are emergent properties of the dynamics produced by our model. The parameter region for highly positively-skewed distributions is the same for both network types. The parameters γ and Δ can be tuned so that all items are attended by everybody or all items are attended by nobody. While not relevant for simulating realistic attendance distributions, these extreme cases help to understand

the model's flexibility.

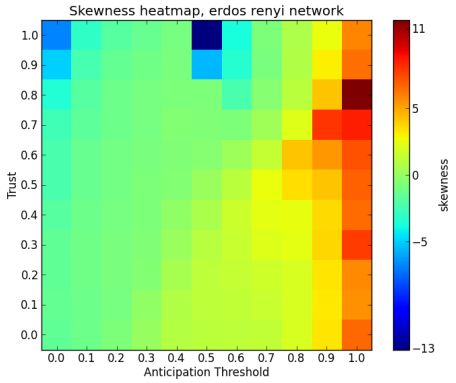


Figure 2: Skewness of the attendance distributions as a function of trust γ and the critical anticipation threshold Δ for Erdős-Rény networks with 500 nodes and 300 simulated items.

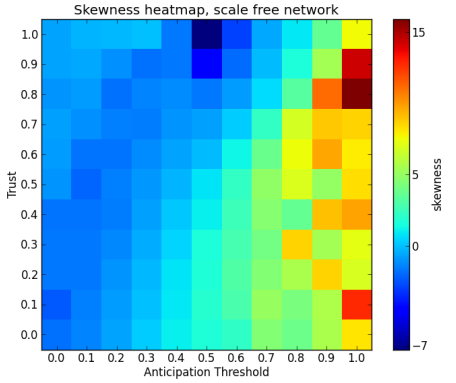


Figure 3: Skewness of the attendance distributions as a function of trust γ and the critical anticipation threshold Δ for power-law networks with 500 nodes and 300 simulated items.

Fitting real data We fit real world recommender data from MovieLens, Netflix and Lastfm with results reported in Fig. (4), Fig. (5), Fig. (6), Fig. (7), and Tab. (1), respectively. The real and simulated distributions are compared using Kullback-Leibler (KL) divergence [29]. We report the mean, median, maximum, and minimum of the simulated and real attendance distributions. Trust γ, anticipation threshold Δ, and anticipation distribution variance σ are reported in figure captions. We also compare the averaged mean degree, maximum degree, minimum degree, and clustering coefficient of the real Lastfm social network and networks obtained to fit the data. Results are reported in Tab. (2) and Fig. (8). Note that thus obtained parameter values can be useful also in real applications where, assuming that our social opinion formation model is valid, one could detect decline of the overall trust value in an online community, for example.

Mathematical analysis. Eq. (6) can be solved analytically. We have $\forall t: \; a(t)+s(t)+d(t) = 1$ with the initial conditions for the first movers $a_0 = \int_\Delta^u f(x)dx$, $s(0) = 1 - a(0)$, and $d(0) = 0$. In the following we use the bra-ket notation $\langle x \rangle$ to represent the average of a quantity x. Standard methods can now be used to arrive at[2]

$$a(t) = \frac{(\tau \langle k \rangle)^{-1} \exp(t/\tau)}{(\alpha + \lambda)\left[\exp(t/\tau) - 1\right] + (\tau \langle k \rangle a_0)^{-1}}. \qquad (7)$$

Here τ is the time scale of the propagation which is defined as

$$\tau = \left(a_0 \alpha \langle k \rangle + \lambda \langle k \rangle\right)^{-1}. \qquad (8)$$

This is similar to the time scale $\tau = (\lambda \langle k \rangle)^{-1}$ in the well known SI Model [38, 6]. Eq.(7) can be very useful in predicting the average behavior of users in a recommender system.

Since Eq. (4) is not accessible to a full analytical solution, we investigate it for the early stage of the dynamics. As-

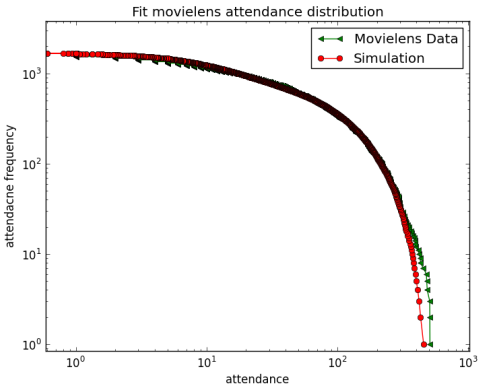


Figure 4: Fit of the MovieLens attendance distribution with trust $\gamma = 0.50$, critical anticipation threshold $\Delta = 0.6$, anticipation distribution variance $\sigma = 0.25$, and power law network with exponent $\delta = 2.25$, 943 nodes, and 1682 simulated objects.

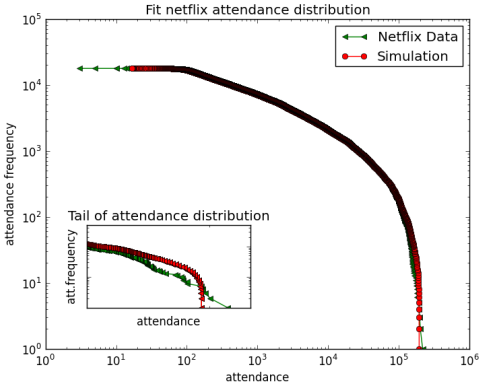


Figure 5: Fit of the Netflix attendance distribution with trust $\gamma = 0.52$, critical anticipation threshold $\Delta = 0.72$, anticipation distribution variance $\sigma = 0.27$, and power law network with exponent $\delta = 2.2$, 480189 nodes, and 17770 simulated objects.

[2]We give here only the solution for $a(t)$ because we are mainly interested in the attendance dynamics.

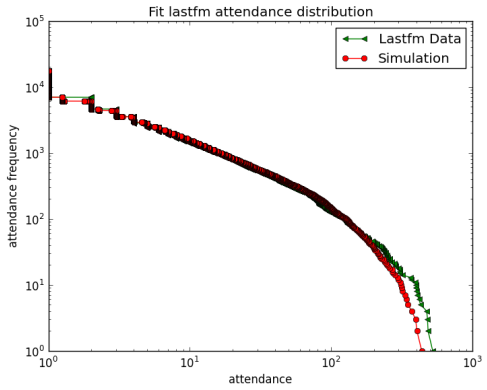


Figure 6: Fit of the Lastfm attendance distribution with trust $\gamma = 0.4$, critical anticipation threshold $\Delta = 0.8$, anticipation distribution variance $\sigma = 0.24$, and real Lastfm user friendship network with 1892 nodes and 17632 simulated objects.

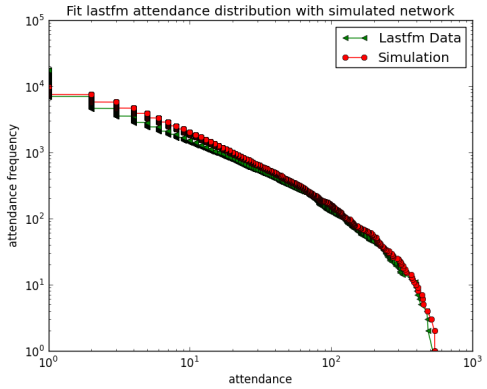


Figure 7: Fit of the Lastfm attendance distribution with trust $\gamma = 0.6$, critical anticipation threshold $\Delta = 0.8$, anticipation distribution variance $\sigma = 0.24$, and power law network with exponent $\delta = 2.25$, 1892 nodes and 17632 simulated objects.

suming $a(0) = a_0 \gg 0$, we can neglect the dynamics of $d(t)$ to obtain

$$\dot{\Omega}(t) = \left(\frac{\langle k^2 \rangle}{\langle k \rangle} - 1 \right) \Omega(t).$$

In addition, Eq. (4) yields

$$\left.\begin{aligned} \dot{a}_k(t) &= \lambda k (1 - a_k(t)) \Omega(t) \\ \dot{s}_k(t) &= -(\alpha + \lambda) k (1 - a_k(t)) \Omega(t) \end{aligned}\right\} \quad (9)$$

Neglecting terms of order $a_k^2(t)$ and summing the solution of $a_k(t)$ over $P(k)$, we get a result for the early spreading stage

$$a(t) = a(0)\Big(1 + \tau\lambda\big(\exp(t/\tau) - 1\big)\Big), \quad (10)$$

with the timescale $\tau = \langle k^2 \rangle / \left[\lambda(\langle k^2 \rangle - \langle k \rangle)\right]$. The obtained time scale τ valid in the early stage of the opinion spreading is clearly dominated by the network heterogeneity. This result is in line with known disease models, e.g., SI,SIR [38, 6].

D	KL	Med	Mean	Max	Min
ML	0.046	27/26	59/60	583/485	1/1
NF	0.030	561/561	5654/5837	232944/193424	3/16
LFM1	0.05	1/1	5.3/5.2	611/503	1/1
LFM2	0.028	1/1	5.3/5.8	611/547	1/1

Table 1: Simulation results. ML: Movielens, NF: Netflix, LFM1: Lastfm with real network, LFM2: Lastfm with simulated network, KL: Kullback-Leibler divergence, Med: Median, Mean, Max: maximal attendance (data/simulated), Min: minimal attendance (data/simulated).

D	$\langle k \rangle$	k_{min}	k_{max}	δ	C
LFM1	13.4	1	119	2.3	0.186
LFM2	12.0	1	118	2.25	0.06

Table 2: Mean, minimum, maximum degree, clustering coefficient C, and estimated exponent δ of the real (LFM1) and simulated (LFM2) social network for the Lastfm data set.

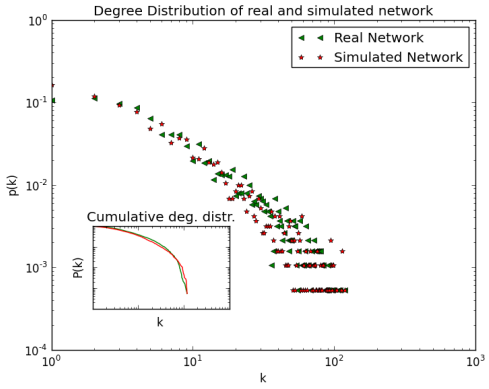


Figure 8: Log-log plot of real (red) and simulated (blue) social network degree distribution $P(k)$ for the Lastfm data set. Inset: plot of the cumulative degree distribution.

We emphasize that Eq.(10) is valuable in predicting users' behavior of a recommender system in an early stage.

5. DISCUSSION

Social influence and our peers are known to form and influence many of our opinions and, ultimately, decisions. We propose here a simple model which is based on heterogeneous agent expectations, a social network, and a formalized social influence mechanism. We analyze the model by numerical simulations and by master equation approach which is particularly suitable to describe the initial phase of the social "contagion". The proposed model is able to generate a wide range of different attendance distributions, including those observed in popular real systems (Netflix, Lastfm, and Movielens). In addition, we showed that these patterns are emergent properties of the dynamics and not imposed by topology of the underlying social network. Of particular interest is the case of Lastfm where the underlying social network is known. Calibrating the observed attendance dis-

tribution against the model then leads not only to social influence parameters but also to the degree distribution of the social network which agrees with that of the true social network.

The Kullback-Leibler distances (KL) for the simulated and real attendance distributions are below 0.05 in all cases, thus demonstrating a good fit. However, the maximum attendances could not be reproduced exactly by the model. One reason may be missing degree correlations in the simulated networks in contrast to real networks where positive degree correlations (so-called degree assortativity) are common. For the Lastfm user friendship network we observe a higher clustering coefficient $C \approx 0.18$ compared to the clustering coefficient $C \approx 0.06$ in the simulated network. To compensate for this, a higher trust parameter γ is needed to fit the real Lastfm attendance distribution with simulated networks.

We are aware that our statistics to validate the model is not complete. But we are confident, that our approach points to a fruitful research direction to understand recommender systems' data as a social driven process.

The proposed model can be a first step towards a data generator to simulate bipartite user-object data with real-world data properties. This could be used to test and validate new recommender algorithms and methods. Future research directions may expand the proposed model to generate ratings within a predefined scale. Moreover, it could be very interesting to investigate the model in the scope of social imitation [36].

6. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, pages 734–749, 2005.

[2] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *Proceeding of the 17th international conference on World Wide Web*, pages 199–208. ACM, 2008.

[3] E. Anderson and L. Salisbury. The formation of market-level expectations and its covariates. *Journal of Consumer Research*, 30(1):115–124, 2003.

[4] J. Arndt. Role of product-related conversations in the diffusion of a new product. *Journal of Marketing Research*, 4(3):291–295, 1967.

[5] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[6] A. Barrat, M. Barthlemy, and A. Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press New York, NY, USA, 2008.

[7] S. Bergamaschi, F. Guerra, and B. Leiba. Information overload. *Internet Computing, IEEE*, 14(6):10–13, 2010.

[8] D. Billsus and M. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, volume 54, page 48, 1998.

[9] A. Birukov, E. Blanzieri, and P. Giorgini. Implicit: An agent-based recommendation system for web search. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 618–624. ACM, 2005.

[10] M. Blattner. B-rank: A top N recommendation algorithm. In *Proceedings of The International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2010)*, volume 1, pages 337–341, Orlando, USA, 2010.

[11] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco, CA, 1998. Morgan Kaufmann.

[12] P. Brusilovsky, A. Kobsa, and W. Nejdl. *The adaptive web: methods and strategies of web personalization*. Springer-Verlag New York Inc, 2007.

[13] G. Caron-Lormier, R. Humphry, D. Bohan, C. Hawes, and P. Thorbek. Asynchronous and synchronous updating in individual-based models. *ecological modelling*, 212(3-4):522–527, 2008.

[14] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proc. ACM SIGIR 99, Workshop Recommender Systems: Algorithms and Evaluation*, 1999.

[15] H. Drachsler, T. Bogers, R. Vuorikari, K. Verbert, E. Duval, N. Manouselis, G. Beham, S. Lindstaedt, H. Stern, M. Friedrich, et al. Issues and considerations regarding sharable data sets for recommender systems in technology enhanced learning. *Procedia Computer Science*, 1(2):2849–2858, 2010.

[16] F. Fouss, A. Pirotte, J. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):355–369, 2007.

[17] F. Fouss, L. Yen, A. Pirotte, and M. Saerens. An experimental investigation of graph kernels on a collaborative recommendation task. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 863–868. IEEE, 2007.

[18] W. Geyer, J. Freyne, B. Mobasher, S. Anand, and C. Dugan. 2nd workshop on recommender systems and the social web. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 379–380. ACM, 2010.

[19] J. Gleeson. High-accuracy approximation of binary-state dynamics on networks. *Physical Review Letters*, 107(6):68701, 2011.

[20] D. Goldberg, B. D. Nichols, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

[21] N. Good, J. Schafer, J. Konstan, A. Brochers, B. Sarwar, J. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proc. Conf. Am. Assoc. Artificial Intelligence (AAAI-99)*, pages 439–446, USA, 1999.

[22] I. Guy, A. Jaimes, P. Agulló, P. Moore, P. Nandy, C. Nastar, and H. Schinzel. Will recommenders kill

search?: recommender systems-an industry perspective. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 7–12. ACM, 2010.

[23] T. Hennig-Thurau, K. Gwinner, G. Walsh, and D. Gremler. Electronic word-of-mouth via consumer-opinion platforms: What motivates consumers to articulate themselves on the Internet? *Journal of Interactive Marketing*, 18(1):38–52, 2004.

[24] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[25] D. Jannach, W. Geyer, J. Freyne, S. Anand, C. Dugan, B. Mobasher, and A. Kobsa. Recommender Systems & the Social Web. In *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009*. ACM, 2009.

[26] K.Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: a constant time collaborative filtering algorithm. *Information Retrieval*, (4):133–151, 2001.

[27] Y. Kim and J. Srivastava. Impact of social influence in e-commerce decision making. *Proceedings of the ninth international conference on Electronic commerce (ICEC)*, pages 293–302, 2007.

[28] J. Konstan, B. Miller, D. Maltz, J. Herlocker, and L. Gordon. Grouplens: Applying collaborative filtering to usenet news. *Comm. ACM*, 40(3):77–87, 1997.

[29] S. Kullback and R. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[30] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[31] M. Mason, R. Dyer, and M. Norton. Neural mechanisms of social influence. *Organizational Behavior and Human Decision Processes*, 110(2):152–159, 2009.

[32] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 492–508, 2004.

[33] P. Massa and B. Bhattacharjee. Using trust in recommender systems: an experimental analysis. *Trust Management*, pages 221–235, 2004.

[34] M.E.J.Newman. Structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[35] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proc. 18h Nat'l Conf. ARtificial Intelligence*, 2002.

[36] Q. Michard and J. Bouchaud. Theory of collective opinion shifts: from smooth trends to abrupt swings. *The European Physical Journal B-Condensed Matter and Complex Systems*, 47(1):151–159, 2005.

[37] B. Mirza, B. Keller, and N. Ramakrishnan. Studying recommendation algorithms by graph analysis. *Journal of Intelligent Information Systems*, 20(2):131–160, 2003.

[38] M. Newman. Spread of epidemic disease on networks. *Physical Review E*, 66(1):016128, 2002.

[39] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174. ACM, 2005.

[40] M. Pazzani and D. Billsus. Content-based recommendation systems. *Lecture Notes Computer Science*, 4321:325–341, 2007.

[41] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netwews. In *Proc. Computer Supported Cooberative Work Conf.*, 1994.

[42] P. Resnick and H. Varian. Recommender systems. *Commun. ACM*, 40:56–58, March 1997.

[43] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[44] M. Richins and T. Root-Shaffer. THE ROLE OF EVOLVEMENT AND OPINION LEADERSHIP IN CONSUMER WORD-OF-MOUTH: AN IMPLICIT MODEL MADE EXPLICIT. *Advances in consumer research*, 15:32–36, 1988.

[45] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM Press.

[46] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684. ACM, 2004.

[47] F. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, 2008.

[48] G. Webb, M. Pazzani, and D. Billsus. Machine learning for user modeling. *User Modeling and User-Adapted Interaction*, 11(1):19–29, 2001.

[49] W. Whyte Jr. The web of word of mouth, 1954.

[50] T. Zhang and V. Iyengar. Recommender systems using linear classifiers. *The Journal of Machine Learning Research*, 2:334, 2002.

[51] Y. Zhang, M. Blattner, and Y. Yu. Heat conduction process on community networks as a recommendation model. *Physical review letters*, 99(15):154301, 2007.

[52] T. Zhou, J. Ren, M. Medo, and Y. Zhang. Bipartite network projection and personal recommendation. *Physical Review E*, 76(4):46115, 2007.

[53] D. Zwillinger and S. Kokoska. *CRC standard probability and statistics tables and formulae*. CRC, 2000.

Effects of Online Recommendations on Consumers' Willingness to Pay

Gediminas Adomavicius
University of Minnesota
Minneapolis, MN
gedas@umn.edu

Jesse Bockstedt
University of Arizona
Tucson, AZ
bockstedt@email.arizona.edu

Shawn Curley
University of Minnesota
Minneapolis, MN
curley@umn.edu

Jingjing Zhang
Indiana University
Bloomington, IN
jjzhang@indiana.edu

ABSTRACT

We present the results of two controlled behavioral studies on the effects of online recommendations on consumers' economic behavior. In the first study, we found strong evidence that participants' willingness to pay was significantly affected by randomly assigned song recommendations, even when controlling for participants' preferences and demographics. In the second study, we presented participants with actual system-generated recommendations that were intentionally perturbed (i.e., significant error was introduced) and observed similar effects on willingness to pay. The results have significant implications for the design and application of recommender systems as well as for e-commerce practice.

1. INTRODUCTION

Recommender systems have become commonplace in online purchasing environments. Much research in information systems and computer science has focused on algorithmic design and improving recommender systems' performance (see Adomavicius & Tuzhilin 2005 for a review). However, little research has explored the impact of recommender systems on consumer behavior from an economic or decision-making perspective. Considering how important recommender systems have become in helping consumers reduce search costs to make purchase decisions, it is necessary to understand how online recommender systems influence purchases.

In this paper, we investigate the relationship between recommender systems and consumers' economic behavior. Drawing on theory from behavioral economics, judgment and decision-making, and marketing, we hypothesize that online recommendations[1] significantly pull a consumer's willingness to pay in the direction of the recommendation. We test our hypotheses using two controlled behavioral experiments on the recommendation and sale of digital songs. In the first study, we find strong evidence that randomly generated recommendations (i.e., not based on user preferences) significantly impact consumers' willingness to pay, even when we control for user preferences for the song, demographic and consumption-related factors, and individual level heterogeneity. In the second study, we extend these results and find strong evidence that these effects still exist with real recommendations generated by a live real-time recommender system. The results of the second study demonstrate that errors in recommendation, a common feature of live recommender systems, can significantly impact the economic behaviors of consumers toward the recommended products.

2. LITERATURE REVIEW AND HYPOTHESES

Behavioral research has indicated that judgments can be constructed upon request and, consequently, are often influenced by elements of the environment. One such influence arises from the use of an anchoring-and-adjustment heuristic (Tversky and Kahneman 1974; see review by Chapman and Johnson 2002), the focus of the current study. Using this heuristic, the decision maker begins with an initial value and adjusts it as needed to arrive at the final judgment. A systematic bias has been observed with this process in that decision makers tend to arrive at a judgment that is skewed toward the initial anchor.

Past studies have largely been performed using tasks for which a verifiable outcome is being judged, leading to a bias measured against an objective performance standard (e.g., see review by Chapman and Johnson 2002). In the recommendation setting, the judgment is a subjective preference and is not verifiable against an objective standard. This aspect of the recommendation setting is one of the task elements illustrated in Figure 1, where accuracy is measured as a comparison between the rating prediction and the consumer's actual rating, a subjective outcome. Also illustrated in Figure 1 is the feedback system involved in the use of recommender systems. Predicted ratings (recommendations) are systematically tied to the consumer's perceptions of products. Therefore, providing consumers with a predicted "system rating" can potentially introduce anchoring biases that significantly influence their subsequent ratings of items.

One of the few papers identified in the mainstream anchoring literature that has looked directly at anchoring effects in preference construction is that of Schkade and Johnson (1989). However, their work studied preferences between abstract, stylized, simple (two-outcome) lotteries. This preference situation is far removed from the more realistic situation that we address in this work. More similar to our setting, Ariely et al. (2003) observed anchoring in bids provided by students participating in auctions of consumer products (e.g., wine, books, chocolates) in a classroom setting. However, participants were not allowed to sample the goods, an issue we address in this study.

[1] In this paper, for ease of exposition, we use the term "recommendations" in a broad sense. Any rating that the consumer receives purportedly from a recommendation system, even if negative (e.g., 1 star on a five-star scale), is termed a recommendation of the system.

Paper presented at the 2012 Decisions@RecSys workshop in conjunction with the 6th ACM conference on Recommender Systems. Copyright © 2012 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

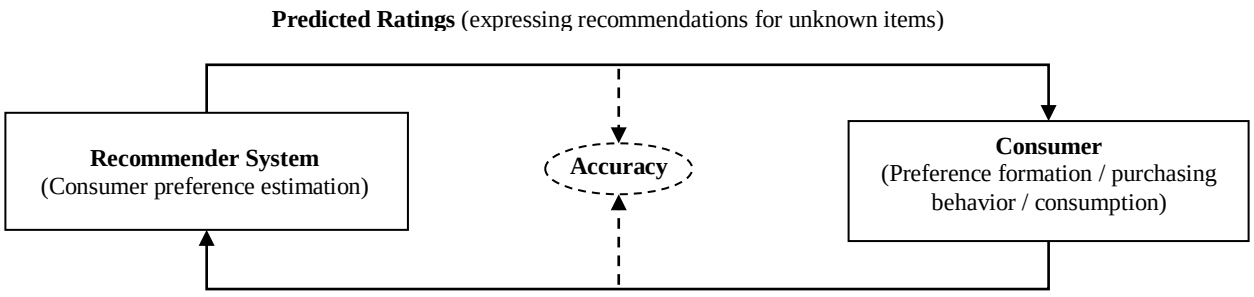


Figure 1. Ratings as part of a feedback loop in consumer-recommender interactions.

Very little research has explored how the cues provided by recommender systems influence online consumer behavior. Cosley et al. (2003) dealt with a related but significantly different anchoring phenomenon in the context of recommender systems. They explored the effects of system-generated recommendations on user re-ratings of movies. They found that users showed high test-retest consistency when being asked to re-rate a movie with no prediction provided. However, when users were asked to re-rate a movie while being shown a "predicted" rating that was altered upward or downward from their original rating by a single fixed amount of one rating point (providing a high or a low anchor), users tended to give higher or lower ratings, respectively (compared to a control group receiving accurate original ratings). This showed that anchoring could affect consumers' ratings based on preference recall, for movies seen in the past and now being evaluated.

Adomavicius et al. (2011) looked at a similar effect in an even more controlled setting, in which the consumer preference ratings for items were elicited at the time of item consumption. Even without a delay between consumption and elicited preference, anchoring effects were observed. The predicted ratings, when perturbed to be higher or lower, affected the consumer ratings to move in the same direction. The effects on consumer ratings are potentially important for a number of reasons, e.g., as identified by Cosley et al. (2003): (1) Biases can contaminate the inputs of the recommender system, reducing its effectiveness. (2) Biases can artificially improve the resulting accuracy, providing a distorted view of the system's performance. (3) Biases might allow agents to manipulate the system so that it operates in their favor. Therefore, it is an important and open research question as to the direct effects of recommendations on consumer behavior.

However, in addition to the preference formation and consumption issues, there is also the purchasing decision of the consumer, as mentioned in Figure 1. Aside from the effects on ratings, there is the important question of the possibility of anchoring effects on economic behavior. Hence, the primary focus of this research is to determine how anchoring effects created by online recommendations impact consumers' economic behavior as measured by their willingness to pay. Based on the prior research, we expect there to be similar effects on economic behavior as observed with consumer ratings and preferences. Specifically, we first hypothesize that recommendations will significantly impact consumers' economic behavior by pulling their willingness to pay in the direction of the recommendation, regardless of the accuracy of the recommendation.

> Hypothesis 1: Participants exposed to randomly generated artificially high (low) recommendations for a product will exhibit a higher (lower) willingness to pay for that product.

A common issue for recommender systems is error (often measured by RMSE) in predicted ratings. This is evidenced by Netflix's recent competition for a better recommendation algorithm with the goal of reducing prediction error by 10% (Bennet and Lanning 2007). If anchoring biases can be generated by recommendations, then accuracy of recommender systems becomes all the more important. Therefore, we wish to explore the potential anchoring effects introduced when real recommendations (i.e., based on the state-of-the-art recommender systems algorithms) are erroneous. We hypothesize that significant errors in real recommendations can have similar effects on consumers' behavior as captured by their willingness to pay for products.

> Hypothesis 2: Participants exposed to a recommendation that contains significant error in an upward (downward) direction will exhibit a higher (lower) willingness to pay for the product.

We test these hypotheses with two controlled behavioral studies, discussed next.

3. STUDY 1: RECOMMENDATIONS AND WILLINGNESS-TO-PAY

Study 1 was designed to test Hypothesis 1 and establish whether or not randomly generated recommendations could significantly impact a consumer's willingness to pay.

3.1. Procedure

Both studies presented in this paper were conducted using the same behavioral research lab at a large public North American university, and participants were recruited from the university's research participant pool. Participants were paid a $10 fee plus a $5 endowment that was used in the experimental procedure (discussed below). Summary statistics on the participant pool for both Study 1 and Study 2 are presented in Table 1. Seven participants were dropped from Study 1 because of response issues, leaving data on 42 participants for analysis.

The experimental procedure for Study 1 consisted of three main tasks, all of which were conducted on a web-based application using personal computers with headphones and dividers between

participants. In the first task, participants were asked to provide ratings for at least 50 popular music songs on a scale from one to five stars with half-star increments. The songs presented for the initial rating task were randomly selected from a pool of 200 popular songs, which was generated by taking the songs ranked in the bottom half of the year-end Billboard 100 charts from 2006 and 2009.[2] For each song, the artist name(s), song title, duration, album name, and a 30-second sample were provided. The objective of the song-rating task was to capture music preferences from the participants so that recommendations could later be generated using a recommendation algorithm (in Study 2 and post-hoc analysis of Study 1, as discussed later).

Table 1 Participant summary statistics.

	Study 1	Study 2
# of Participants (n)	42	55
Average Age (years)	21.5 (1.95)	22.9 (2.44)
Gender	28 Female, 14 Male	31 Female, 24 Male
Prior experience with recommender systems	50% (21/42)	47.3% (26/55)
Student Level	36 undergrad, 6 grad	27 undergrad, 25 grad, 3 other
Buy new music at least once a month	66.7% (28/42)	63.6% (35/55)
Own more than 1000 songs	50% (21/42)	47.3% (26/55)

In the second task, a different list of songs was presented (with the same information for each song as in the first task) from the same set of 200 songs. For each song, the participant was asked whether or not they owned the song. Songs that were owned were excluded from the third task, in which willingness-to-pay judgments were obtained. When the participants identified at least 40 songs that they did not own, the third task was initiated.

In the third main task of Study 1, participants completed a within-subjects experiment where the treatment was the star rating of the song recommendation and the dependent variable was willingness to pay for the songs. In the experiment, participants were presented with 40 songs that they did not own, which included a star rating recommendation, artist name(s), song title, duration, album name, and a 30 second sample for each song. Ten of the 40 songs were presented with a randomly generated low recommendation between one and two stars (drawn from a uniform distribution; all recommendations were presented with a one decimal place precision, e.g., 1.3 stars), ten were presented with a randomly generated high recommendation between four and five stars, ten were presented with a randomly generated mid-range recommendation between 2.5 and 3.5 stars, and ten were presented with no recommendation to act as a control. The 30 songs presented with recommendations were randomly ordered, and the 10 control songs were presented last.

To capture willingness to pay, we employed the incentive-compatible Becker-DeGroot-Marschack method (BDM) commonly used in experimental economics (Becker et al. 1984). For each song presented during the third task of the study, participants were asked to declare a price they were willing to pay between zero and 99 cents. Participants were informed that five songs selected at random at the end of the study would be assigned random prices, based on a uniform distribution, between one and 99 cents. For each of these five songs, the participant was required to purchase the song using money from their $5 endowment at the randomly assigned price if it was equal to or below their declared willingness to pay. Participants were presented with a detailed explanation of the BDM method so that they understood that the procedure incentivizes accurate reporting of their prices, and were required to take a short quiz on the method and endowment distribution before starting the study.

At the conclusion of the study, they completed a short survey collecting demographic and other individual information for use in the analyses. The participation fee and the endowment minus fees paid for the required purchases were distributed to participants in cash. MP3 versions of the songs purchased by participants were "gifted" to them through Amazon.com approximately within 12 hours after the study was concluded.

3.2. Analysis and Results

We start by presenting a plot of the aggregate means of willingness to pay for each of the treatment groups in Figure 2. Note that, although there were three treatment groups, the actual ratings shown to the participants were randomly assigned star ratings from within the corresponding treatment group range (low: 1.0-2.0 stars, mid: 2.5-3.5 stars, high: 4.0-5.0 stars).

As an initial analysis, we performed a repeated measure ANOVA, as shown in Table 2, demonstrating a statistically significant effect of the shown rating on willingness to pay. Since the overall treatment effect was significant, we followed with pair-wise contrasts using t-tests across treatment levels and against the control group as shown in Table 3. All three treatment conditions significantly differed, showing a clear, positive effect of the treatment on economic behavior.

To provide additional depth for our analysis, we used a panel data regression model to explore the relationship between the shown star rating (continuous variable) and willingness to pay, while controlling for participant level factors. A Hausman test was conducted, and a random effects model was deemed appropriate, which also allowed us to account for participant level covariates in the analysis. The dependent variable, i.e., willingness to pay, was measured on an integer scale between 0 and 99 and skewed toward the lower end of the scale. This is representative of typical count data; therefore, a Poisson regression was used (overdispersion was not an issue). The main independent variable was the shown star rating of the recommendation, which was continuous between one and five stars. Control variables for several demographic and consumer-related factors were included, which were captured in the survey at the end of the study. Additionally, we controlled for the participants' preferences by calculating an actual predicted star rating recommendation for each song (on a 5 star scale with one decimal precision), post hoc, using the popular and widely-used item-based collaborative

[2] The Billboard 100 provides a list of popular songs released in each year. The top half of each year's list was not used to reduce the number of songs in our database that participants would already own.

filtering algorithm (IBCF) (Sarwar et al. 2001).[3] By including this predicted rating (which was not shown to the participant during the study) in the analysis, we are able to determine if the random recommendations had an impact on willingness to pay above and beyond the participant's predicted preferences.

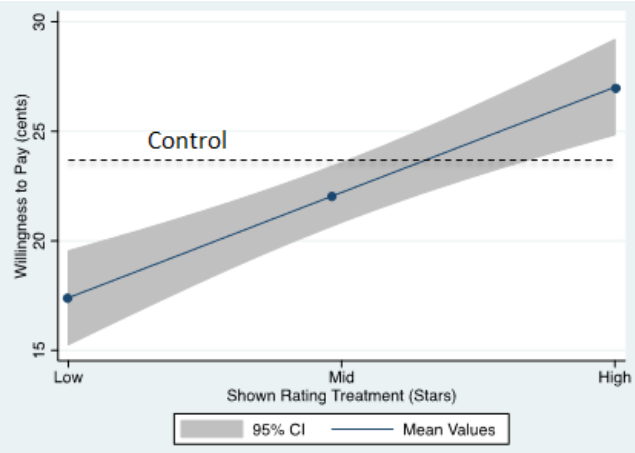


Figure 2. Study 1 treatment means.

Table 2. Study 1 repeated measures ANOVA.

	Partial Sum of Squares	Degrees of Freedom	Mean Square	F Statistic	P value
Participant	396744.78	41	9676.70		
Treatment Level	24469.41	2	12234.70	42.27	<0.000
Residual	346142.41	1196	289.42		
Total	762747.40	1239	615.62		

Table 3. Comparison of aggregate treatment group means with *t*-tests.

	Control	*Low*	*Mid*
Low (1-2 Star)	4.436***		
Mid (2.5-3.5 Star)	0.555	4.075***	
High (4-5 Star)	1.138	5.501***	1.723**

* p<0.1, ** p<0.05, *** p <0.01
2-tailed t-test for *Control* vs. *Mid*, all else 1-tailed.

The resulting Poisson regression model is shown below, where WTP_{ij} is the reported willingness to pay for participant i on song j, $ShownRating_{ij}$ is the recommendation star rating shown to participant i for song j, $PredictedRating_{ij}$ is the predicted recommendation star rating for participant i on song j, and $\mathbf{Controls}_i$ is a vector of demographic and consumer-related variables for participant i. The controls included in the model were gender (binary), age (integer), school level (undergrad yes/no binary), whether they have prior experience with recommendation systems (yes/no binary), preference ratings (interval five point scale) for the music genres country, rock, hip hop, and pop, the number of songs owned (interval five point scale), frequency of music purchases (interval five point scale), whether they thought recommendations in the study were accurate (interval five point scale), and whether they thought the recommendations were useful (interval five point scale). The composite error term $(u_i + \varepsilon_{ij})$ includes the individual participant effect u_i and the standard disturbance term ε_{ij}.

$$\log(WTP_{ij}) = b_0 + b_1(ShownRating_{ij}) + b_2(PredictedRating_{ij}) + \mathbf{b_3}(\mathbf{Controls}_i) + u_i + \varepsilon_{ij}$$

The results of the regression are shown in Table 4. Note that the control observations were not included, since they had null values for the main dependent variable *ShownRating*.

The results of our analysis for Study 1 provide strong support for Hypothesis 1 and demonstrate clearly that there is a significant effect of recommendations on consumers' economic behavior. Specifically, we have shown that even randomly generated recommendations with no basis on user preferences can impact consumers' perceptions of a product and, thus, their willingness to pay. The regression analysis goes further and controls for participant level factors and, most importantly, the participant's predicted preferences for the product being recommended. As can be seen in Table 4, after controlling for all these factors, a one unit change in the shown rating results in a 0.168 change (in the same direction) in the log of the expected willingness to pay (in cents). As an example, assuming a consumer has a willingness to pay of \$0.50 for a specific song and is given a recommendation, increasing the recommendation star rating by one star would increase the consumer's willingness to pay to \$0.59.

Table 4. Study 1 regression results

Dependent Variable: log(Willingness to Pay)		
Variable	**Coefficient**	**Std. Error**
ShownRating	0.168***	0.004
PredictedRating	0.323***	0.015
Controls		
male	-0.636**	0.289
undergrad	-0.142	0.642
age	-0.105	0.119
usedRecSys	-0.836**	0.319
country	0.103	0.108
rock	0.125	0.157
hiphop	0.152	0.132
pop	0.157	0.156
recomUseful	-0.374	0.255
recomAccurate	0.414*	0.217
buyingFreq	-0.180	0.175
songsOwned	-0.407*	0.223
constant	4.437	3.414
Number of Obs.	1240	
Number of Participants	42	
Log-likelihood	-9983.3312	
Wald Chi-Square Statistic	1566.34	
(p-value)	(0.0000)	

* p<0.1, ** p<0.05, *** p <0.01

4. STUDY 2: ERRORS IN RECOMMENDATIONS

The goal of Study 2 was to extend the results of Study 1 by testing Hypothesis 2 and exploring the impact of significant error in true

[3] Several recommendation algorithms were evaluated based on the Study 1 training data, and IBCF was selected for us in both studies because it had the highest predictive accuracy.

recommendations on consumers' willingness to pay. As discussed below, the design of this study is intended to test for similar effects as Study 1, but in a more realistic setting with recommender-system-generated, real-time recommendations..

4.1. Procedure

Participants in Study 2 used the same facilities and were recruited from the same pool as in Study 1; however, there was no overlap in participants across the two studies. The same participation fee and endowment used in Study 1 was provided to participants in Study 2. 15 participants were removed from the analysis in Study 2 because of issues in their responses, leaving data on 55 participants for analysis.

Study 2 was also a within-subjects design with perturbation of the recommendation star rating as the treatment and willingness to pay as the dependent variable. The main tasks for Study 2 were virtually identical to those in Study 1. The only differences between the studies were the treatments and the process for assigning stimuli to the participants in the recommendation task of the study. In Study 2, all participants completed the initial song-rating and song ownership tasks as in Study 1. Next, real song recommendations were calculated based on the participants' preferences, which were then perturbed (i.e., excess error was introduced to each recommendation) to generate the shown recommendation ratings. In other words, unlike Study 1 in which random recommendations were presented to participants, in Study 2 participants were presented with perturbed versions of their actual personalized recommendations. Perturbations of -1.5 stars, -1 star, -0.5 stars, 0 stars, +0.5 stars, +1 star, and +1.5 stars were added to the actual recommendations, representing seven treatment levels. The perturbed recommendation shown to the participant was constrained to be between one and five stars, therefore perturbations were pseudo-randomly assigned to ensure that the sum of the actual recommendation and the perturbation would fit within the allowed rating scale. The recommendations were calculated using the item-based collaborative filtering (IBCF) algorithm (Sarwar et al. 2001), and the ratings data from Study 1 was used as training data.

Each participant was presented with 35 perturbed, personalized song recommendations, five from each of the seven treatment levels. The song recommendations were presented in a random order. Participants were asked to provide their willingness to pay for each song, which was captured using the same BDM technique as in Study 1. The final survey, payouts, and song distribution were also conducted in the same manner as in Study 1.

4.2. Analysis and Results

For Study 2, we focus on the regression analysis to determine the relationship between error in a recommendation and willingness to pay. We follow a similar approach as in Study 1 and model this relationship using a Poisson random effects regression model. The distribution of willingness to pay data in Study 2 was similar to that of Study 1, overdispersion was not an issue, and the results of a Hausman test for fixed versus random effects suggested that a random effects model was appropriate. We control for the participants' preferences using the predicted rating for each song in the study (i.e., the recommendation rating prior to perturbation), which was calculated using the IBCF algorithm. Furthermore, the same set of control variables used in Study 1 was included in our regression model for Study 2. The resulting regression model is presented below, where the main difference from the model used in Study 1 is the inclusion of $Perturbation_{ij}$ (i.e., the error introduced for the recommendation of song j to participant i) as the main independent variable. The results are presented in Table 5.

$$\log(WTP_{ij}) = b_0 + b_1(Perturbation_{ij}) + b_2(PredictedRating_{ij}) + \mathbf{b_3}(\mathbf{Controls}_i) + u_i + \varepsilon_{ij}$$

The results of Study 2 provide strong support for Hypothesis 2 and extend the results of Study 1 in two important ways. First, Study 2 provides more realism to the analysis, since it utilizes real recommendations generated using an actual real-time recommender system. Second, rather than randomly assigning recommendations as in Study 1, in Study 2 the recommendations presented to participants were calculated based on their preferences and then perturbed to introduce realistic levels of system error. Thus, considering the fact that all recommender systems have some level of error in their recommendations, Study 2 contributes by demonstrating the potential impact of these errors. As seen in Table 5, while controlling for preferences and other factors, a one unit perturbation in the actual rating results in a 0.115 change in the log of the participant's willingness to pay. As an example, assuming a consumer has a willingness to pay of $0.50 for a given song, perturbing the system's recommendation positively by one star would increase the consumer's willingness to pay to $0.56.

Table 5. Study 2 regression results.

Dependent Variable: log(Willingness to Pay)		
Variable	**Coefficient**	**Std. Error**
Perturbation	0.115***	0.005
PredictedRating	0.483***	0.012
Controls		
male	-0.045	0.254
undergrad	-0.092	0.293
age	-0.002	0.053
usedRecSys	0.379	0.253
country	-0.056	0.129
rock	-0.132	0.112
hiphop	0.0137	0.108
pop	-0.035	0.124
recomUseful	0.203*	0.112
recomAccurate	0.060	0.161
buyingFreq	0.276**	0.128
songsOwned	-0.036	0.156
constant	0.548	1.623
Number of Obs.	1925	
Number of Participants	55	
Log-likelihood	-16630.547	
Wald Chi-Square Statistic	2374.72	
(p-value)	(0.0000)	

* p<0.1, ** p<0.05, *** p <0.01

5. CONCLUSIONS

Study 1 provided strong evidence that willingness to pay can be affected by online recommendations through a randomized trial design. Participants presented with random recommendations were influenced even when controlling for demographic factors and general preferences. Study 2 extended these results to demonstrate that the same effects exist for real recommendations that contain errors, which were calculated using the state-of-the-art recommendation algorithms used in practice.

There are significant implications of the results presented. First, the results raise new issues on the design of recommender systems. If recommender systems can generate biases in consumer decision-making, do the algorithms need to be adjusted to compensate for such biases? Furthermore, since recommender systems use a feedback loop based on consumer purchase decisions, do recommender systems need to be calibrated to handle biased input? Second, biases in decision-making based on online recommendations can potentially be used to the advantage of e-commerce companies, and retailers can potentially become more strategic in their use of recommender systems as a means of increasing profit and marketing to consumers. Third, consumers may need to become more cognizant of the potential decision making biases introduced through online recommendations. Just as savvy consumers understand the impacts of advertising, discounting, and pricing strategies, they may also need to consider the potential impact of recommendations on their purchasing decisions.

6. ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation grant IIS-0546443.

REFERENCES

[1] Adomavicius, G., Bockstedt, J., Curley, S., and Zhang, J. 2011. Recommender Systems, Consumer Preferences, and Anchoring Effects. *Proceedings of the RecSys 2011 Workshop on Human Decision Making in Recommender Systems (Decisions@RecSys 2011)*, Chicago IL, October 27, pp. 35-42.

[2] Adomavicius, G. and Tuzhilin, A. 2005. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17 (6) pp. 734-749.

[3] Ariely, D., Lewenstein, G., and Prelec, D. 2003. "Coherent arbitrariness": Stable demand curves without stable preferences, *Quarterly Journal of Economics* (118), pp. 73-105.

[4] Becker G.M., DeGroot M.H., and Marschak J. 1964. Measuring utility by a single-response sequential method. *Behavioral Science*, 9 (3) pp. 226–32.

[5] Bennet, J. and Lanning, S. 2007. The Netflix Prize. *KDD Cup and Workshop*. [www.netflixprize.com].

[6] Chapman, G. and Johnson, E. 2002. Incorporating the irrelevant: anchors in judgments of belief and value. *Heuristics and Biases: The Psychology of Intuitive Judgment*, T. Gilovich, D. Griffin and D. Kahneman (eds.), Cambridge University Press, Cambridge, pp. 120-138.

[7] Cosley, D., Lam, S., Albert, I., Konstan, J.A., and Riedl, J. 2003. Is seeing believing? How recommender interfaces affect users' opinions. *CHI 2003 Conference*, Fort Lauderdale FL.

[8] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2001. Item-based collaborative filtering algorithms. *10th Annual World Wide Web Conference (WWW10)*, May 1-5, Hong Kong.

[9] Schkade, D.A. and Johnson, E.J. 1989. Cognitive processes in preference reversals. *Organizational Behavior and Human Decision Processes*, (44), pp. 203-231.

[10] Tversky, A., and Kahneman, D. 1974. Judgment under uncertainty: Heuristics and biases. *Science*, (185), pp. 1124-1131.