Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Case-studies on exploiting explicit customer requirements in recommender systems

Markus Zanker[1] and Markus Jessenitschnig[1,2]
[1] *University Klagenfurt, Austria*
email: firstname.secondname@uni-klu.ac.at
[2] *E-Tourism Competence Center Austria (ECCA)*

June 15, 2008

Abstract. Recommender Systems (RS) suggest useful and interesting items to users in order to increase user satisfaction and online conversion rates. They typically exploit explicit or implicit user feedback such as ratings, buying records or clickstream data and apply statistical methods to derive recommendations.

This paper focuses on explicitly formulated customer requirements as the sole type of user feedback. Its contribution lies in comparing different techniques such as knowledge- and utility-based methods, collaborative filtering, association rule mining as well as hybrid variants when user models consist solely of explicit customer requirements. We examine how this type of user feedback can be exploited for personalization in e-commerce scenarios. Furthermore, examples of actual online shops are developed where such contextual user information is available, demonstrating how more efficient RS configurations can be implemented. Results indicate that, especially for new users, explicit customer requirements are a useful source of feedback for personalization and hybrid configurations of collaborative and knowledge-based techniques achieve best results.

Keywords: hybrid recommender systems, comparative evaluation, electronic commerce, cold-start recommendation problem

1. Introduction

Recommendation technologies have become popular for providing interactivity and personalization within e-commerce platforms over the last decade. Recommender systems (RS) exploit observations about users, such as ratings of items, transaction and clickstream data or explicitly formulated requirements, and derive - based on their reasoning mechanisms - items that might be of particular interest. However, providing meaningful advice and recommendations to anonymous or first-time users still remains a major challenge (known as *cold start* problem for new users). The problem is aggravated by commercial situations where users rarely make multiple purchases within the same product category (e.g. digital cameras or insurances) or where users are anonymous.

So far the following strategies have been proposed to cope with users that are *new* to the system:

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

1. Require users to provide ratings on a set of products. For instance Jester - a joke recommender system - asks users to rate a specific set of jokes as part of the registration process (Goldberg et al., 2001).

2. Collect clickstream information such as pageviews and interpret them as binary ratings. Such an exploitation of navigational patterns does not impose additional effort on users. Several studies such as (Mobasher et al., 2000; Pierrakos et al., 2003) showed that clickstream data can be a valuable substitute for explicit user ratings.

3. Infer users' interest from content information of pages that attracted their interest. Thus in addition to clickstream data, the content of a webpage is used for building user-profiles of first-time visitors (Balabanovic and Shoham, 1997; Wasfi, 1999; Schein et al., 2002).

4. Elicit requirements from users by asking them explicitly. This strategy is typically followed by conversational recommender systems based on knowledge- or utility-based reasoning mechanisms and requires a considerable knowledge-engineering effort when setting up the system (Burke, 2000a; Shimazu, 2001; Ricci et al., 2003; Jannach, 2004; Rafter and Smyth, 2005).

This article focuses on the fourth strategy, namely the explicit elicitation of user requirements. Many shopping portals offer form based dialogues where users can explicitly state what they are looking for or what they are interested in. Comparable to a sales assistant who welcomes and helps customers when they enter a store, parametric search masks and advisory dialogues interact with users and provide them with information and links of interest depending on their reasoning capabilities and the user's situational needs. For instance, a user might be looking for a gift for a friend or require an appropriate wine for a romantic dinner. Exploiting this contextual information in a collaborative filtering system by segmenting user ratings on criteria such as who accompanies the user when watching the movie or where it is seen - at home or in a theater - can significantly improve prediction quality of recommendation algorithms as shown by (Adomavicius et al., 2005). Consequently, explicit requirements like 'the user is looking for a cigar as a present for a friend' or 'she[1] has only moderate smoking experience and some budget restrictions' should make an even bigger difference.

[1] We refer to the user in a unique gender for readability purposes.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Based on data from real-world e-commerce shops and fielded conversational recommender systems for fine Cuban cigars and consumer electronics, different reasoning mechanisms and recommendation paradigms (such as knowledge- and utility-based RS, association rule mining and collaborative filtering) are applied on user models that solely consist of explicit customer requirements elicited via a form-based dialogue. The paper's contribution thus lies in a comparative evaluation of different recommendation techniques when solely applied to explicit user requirements and in the development of cases for identifying efficient hybrid algorithm variants. It explores all weighted and cascaded combinations of the aforementioned recommendation techniques and proposes a hybridization variant that considerably improves the accuracy on sample datasets. As a result it concludes that explicit customer requirements are a valuable type of user feedback for recommender systems, especially for new users entering the system. Furthermore, traditional user-to-user collaborative filtering outperformed comparable techniques when exploiting explicit customer requirements as user feedback. Finally, weighted and switching hybridization strategies on collaborative filtering and knowledge-based recommendation achieve additional improvements.

The following section summarizes related and background work and gives details on the recommendation techniques presented in Section 3. Furthermore, the case study of the evaluation is developed in Section 4. Section 5 validates the findings by accommodating two additional evaluation sets. A discussion on the results and findings as well as an outlook on future work finalizes this article.

2. Related Work

This section gives a short outline of related research on recommendation systems in the field of e-commerce. Since their early beginnings, the GroupLens research group at the University of Minnesota has been one of the pioneers in the field of recommender systems (Resnick et al., 1994). Starting with recommendations for online news (Konstan et al., 1997) and websites (Balabanovic, 1997), the research community quickly turned to movie recommendations as one of their primary application areas. The MovieLens system allows users to choose among thousands of films in order to rate them. Users are rewarded with novel movie recommendations from cineastes with similar tastes by matching their preferences with ratings of other users (Herlocker et al., 1999). The worldwide online bookseller *amazon.com* was among the first to adopt collaborative filtering commercially (Linden et al., 2003) and due to

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

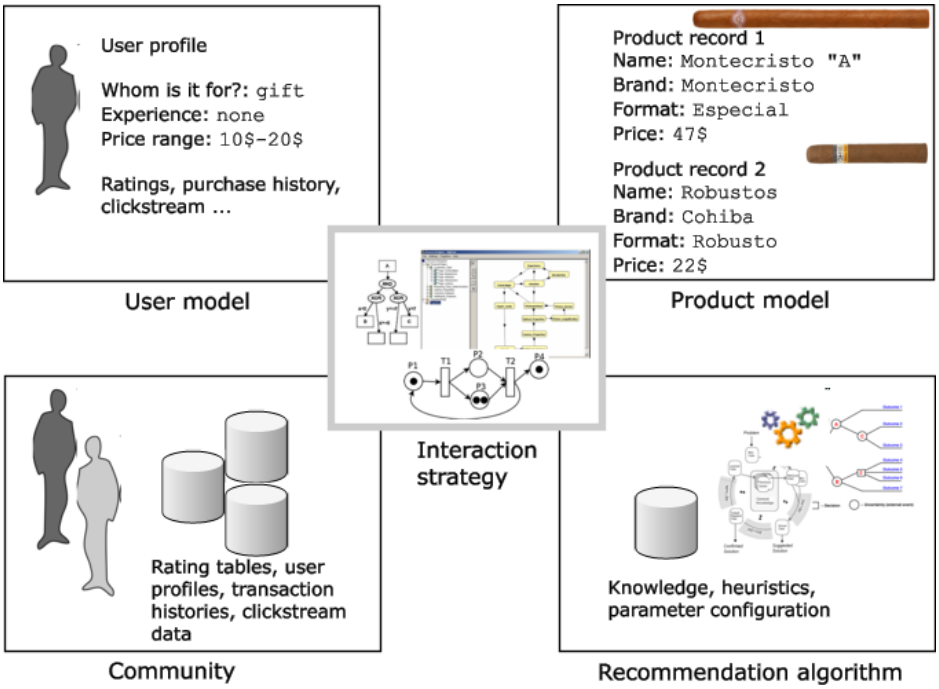

Figure 1. Building blocks of recommender systems

Amazon's overwhelming success became the most prominent example of the application of recommendation technology to e-commerce. See (Goy et al., 2007) for a discussion on personalization in e-commerce applications and (Riedl et al., 2002) for a hands-on overview on successful commercial examples. In research the technology has been expanded to a variety of application domains: music (Shardanand and Maes, 1995), restaurants (Burke, 2002), TV guides (Smyth and Cotter, 2001) or tourism destinations (Ricci and Werthner, 2002; Ricci, 2002) to name a few. It is not surprising that due to the public availability of datasets like MovieLens or EachMovie (Herlocker et al., 2004), most of the research concentrates on developing or optimizing techniques for deriving accurate recommendations in domains with tens of thousands of product items (e.g. movies, books or music), a large body of users and a two-digit number of ratings per user. However, in situations where mostly new or anonymous users interact with the system additional datasets are required for research and evaluation (Zanker et al., 2007).

Figure 1 illustrates the fundamental building blocks of recommender systems: a user model, community, product model, recommendation algorithm and interaction style. These different aspects in the application

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

of recommender systems are consequently discussed and references are given.

User model: *What does the RS know about the current user?*

All Recommender systems require a user model in order to personalize their recommendations and acquire them by preference elicitation and observance of users' behavior. (McGinty and Smyth, 2006) identified the following types of feedback that can be acquired from users: preference for an item a vs. an item b, critique on a presented item, ratings and general preference values. These feedback types differ in terms of their explicitness, validity and acquisition costs. While preferences between items can be simply observed by evaluating click-stream data, users need to be explicitly asked about general preferences and requirements or critiques. The latter are typically represented as sets of constraints, see (Linden et al., 1997; Pu and Faltings, 2004). Pure collaborative filtering systems work solely with user ratings; either on a multi-point Likert or a simple binary scale (i.e. likes/dislikes) (Herlocker et al., 1999). In commercial environments, typically implicit ratings in the form of binary transaction data are collected, signifying wether a user bought a product or showed any interest for instance by accessing a page with more detailed product information. Furthermore, more abstract information such as demographic and personal data as well as concrete needs and requirements might be derived from the user through interactive advisory dialogues.

Community: *What effect does the existence of other user models have on the recommendations for a specific user?*

With respect to community, the distinction must be made between learning-based recommendation methods that rely on neighboring users as the primary source of knowledge for computing recommendations and non learning-based approaches that solely exploit product or means-end knowledge. Collaborative and demographic filtering determine a user's neighborhood based on similar ratings or demographic user characteristics such as age, profession or educational background respectively. Items of interest can be derived based on the ratings or purchases of their nearest neighbors. Consequently, systems with large community bases typically improve over their lifetime. As a tradeoff, community-based algorithms suffer from cold-start problems, when new users access the system or new items that have not been rated yet are introduced to the product base.

Regarding the implementation, memory- and model-based algorithms exist (Breese et al., 1998). Memory-based approaches determine the nearest neighbors for the current user at runtime similarly to user-based

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

collaborative filtering (Herlocker et al., 1999). However, when the user base is large, model-based algorithms are more resource-efficient. They precompute a predictive model that saves computation time during user interaction. (Sarwar et al., 2001) proposed item-based collaborative filtering, where similarities between items are derived from user ratings: for each pair of items all users who rated both of them are identified and a similarity measure such as the cosine of the angle between both rating vectors is applied. When a new user shows interest in one item, the most similar items to this reference item are then retrieved and proposed (Linden et al., 2003). Another variant of a model-based approach to exploit community data is the application of association rule mining for generating product recommendations. It identifies functional dependencies within sets of items (Agrawal et al., 1993). Therefore, association rules can be mined from past user transactions and utilized for predicting recommendable items (Sarwar et al., 2000; Demiriz, 2004). The experiments in Section 4 apply association rule mining as well as item- and user-based collaborative filtering on explicit user requirements and evaluate them.

Product Model: *What does the RS know about products?*

The number of features represented in a product model is strongly influenced by the choice of the recommendation strategy and the domain. While collaborative filtering techniques require solely a unique identifier for each product instance, content- and knowledge-based approaches need additional product descriptions. Depending on the domain, full-text descriptions like news articles (Balabanovic and Shoham, 1997) or structured product representations in the form of attribute-value pairs such as brand, format or price in case of cigars will be part of a product model. Consequently, the availability of product knowledge extends the capabilities of recommender systems (Balabanovic and Shoham, 1997). Regardless of the commercial purpose of a platform, the availability of product knowledge is determined by the domain and the effort invested in its acquisition and maintenance.

Recommendation algorithm: *Which reasoning paradigm does the RS follow?*

Burke (Burke, 2002) distinguished between five different archetypes of recommender systems. Two of them, collaborative and demographic filtering using community characteristics as previously discussed. Contrastingly, the three other paradigms, namely content-, knowledge- and utility-based RS exploit product models. Content-based filtering builds rich user models from the characteristics of items that are of interest to the user. One of the earliest application scenarios was recommend-

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

ing interesting webpages and news (Balabanovic and Shoham, 1997). When recommendable items are represented as a term vector, the user model may contain a vector of term categories and the user's degrees of interest in them. Items of interest are then retrieved based on similarities between term vectors of items and the user model representation. Knowledge- and utility-based RS rely on explicit knowledge that relates user and product models. For instance the Findme system (Burke, 2000b) offers a conversational interface that allows users to formulate a critique on a specific item and retrieves more appropriate alternatives based on similarity measures. For instance, if the user asks for a cigar that has more prestige than product a, then the system would search for models with a well-known brand name and a noble packaging. As can be seen from this example, deep product knowledge comes into play. For example:

- which product model features impact higher-level concepts like *prestige*,
- which similarity measures should be selected or
- how can trade-off decisions (e.g. larger size but also more smoking experience required) be handled.

One of the first systems based on such a candidate/critique interaction style was the *Automated Travel Assistant* presented by (Linden et al., 1997). Such critique systems have been further developed to consider user's critiquing history (Reilly et al., 2005), learn compound critiques (Reilly et al., 2007) or introduce adaptive suggestions of options to elicit critiques (Viappiani et al., 2006).

Ardissono and Goy also employ multi-attribute utility definitions to tailor the interaction in web stores and suggest goods to users (Ardissono and Goy, 2000). They assume abstract properties such as *quality* or *ease of use* and compute an overall matching value between a user's preferences and item characterizations.

(Ricci and Werthner, 2002) present Nutking, a case-based recommendation system for travel planning, which on the one hand retrieves items based on their similarity to the user's query and on the other hand exploits community knowledge (i.e. stored cases or travel plans) to propose bundles of different types, e.g. accommodations, leisure and sporting activities. (Smyth, 2007) provides a comprehensive discussion on further examples of case-based and conversational recommendation systems.

(Jannach, 2004) proposed a domain-independent knowledge-based sales advisory system that evaluates constraints in the form of *if* ...

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

then ... filter conditions. An example for such an implication in classical logic is: *if the user requires a gift for a friend then propose only prestigious brands.* Here, dependencies between user requirements and product features are explicitly represented. Hence, knowledge-based RS exploit domain specific means-end knowledge for making personalized product propositions. Consequently, they rely on heuristics, business rules or constraints that need to be acquired from domain experts during the setup phase of a system. This is one of the main differences to critique systems as discussed before, because they dynamically acquire constraints as preference information during user interaction. Thus, the basic idea of a knowledge-based recommender like (Jannach, 2004) is to model and automate the behavior of experienced sales agents in routine sales advisory tasks and build cost-efficient and easily accessible self-service applications for customers. For instance in domains like financial services, where agents may be held liable for the advice they are giving, automated and quality-assured sales dialogues have been successfully deployed (Felfernig et al., 2006).

In such knowledge-based systems trade-off decisions between conflicting constraints are handled by distinguishing between hard constraints that are always required and soft constraints that might be relaxed to avoid empty result sets. See (Mirzadeh et al., 2004; Jannach, 2006a; Jannach, 2006b) for discussions of relaxation strategies for interactive recommender systems.

As each of these pure types of recommendation paradigms have their pros and cons, a variety of hybrid algorithm variants has been proposed. (Burke, 2002) provides an encompassing survey of hybrid recommender systems. Furthermore, in this issue (Berkovsky et al., 2008) present their work on mediating between user-model representations of different recommendation strategies.

Interaction strategy: *What design does the user interaction of a RS have?*

Although the interaction strategy partly depends on the type of feedback collected from users, it must still be seen as a separate aspect of its own. Implicitly collected information exploiting click-stream and transaction data doesn't lead to additional effort for users. However, explicit rating of tens of items would signify a considerable effort that users, who are looking for the best deal on a consumer product, might not be willing to take. In addition, the validity of an explicit rating largely depends on the user's capability to do so: e.g., judging the haptic appearance or subjectively assessing an item's usability based on its technical parameters and general domain knowledge. This varies strongly between product domains. For instance, in the entertainment

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

domain (movies or music) users can often quickly decide if they like something or not, in contrast to more technical and/or complex domains, such as consumer electronics or financial services, where users may not have an idea of what they prefer (Felfernig et al., 2006). This argument motivates *example critiquing* interaction, too. (Linden et al., 1997) states that it must not be assumed that the full preference information can be provided as input to interactive problem solving in complex domains. Therefore the user model information is inferred over multiple interactions in such critiquing systems.

Coming back to the guiding example of cigars: An experienced sales agent typically does not simply sell a cigar that is currently on sale, but would firstly try to find out about the customer's needs. Therefore, questions like *"how much smoking experience do you have?"* or *"do you already posses a humidor?"* are prerequisites for a human sales agent to personalize her customer advice. In addition, the sales agent will already offer different possible replies to the customer, for instance *"whom is the cigar for? - is it for yourself or do you need it as a gift?"*. These structured sales conversations can easily be modeled and automated as form-based dialogues on the web. This is exactly the basic idea of the *Advisor Suite* system (Jannach, 2004) that is employed here for generating knowledge-based recommendations. It is a conversational recommender system shell, that provides a graphical knowledge acquisition environment for rapid application development (Jannach and Kreutler, 2004). This way, the system enables e-tailers to define their best-practice sales dialogues for different product categories on their own. The system differs from the approach presented in (Elzer et al., 1994) by clearly focusing on online sales situations. Elzer et al. worked on the recognition and exploitation of user preferences in natural language consultation systems as well as on the generation of tailored responses (Chu-Carrol and Carberry, 1994; Harvey et al., 2005). *Advisor Suite* does not understand natural language input. All potential user preferences have to be already known at setup time. Therefore all answering options to questions must be predefined by a domain expert. However, the dialogues are still adaptive to the user in the sense of adaptive selection of communication paths or personalized hints and explanations (Jannach and Kreutler, 2005). Adaptive guidance through large information bases was also proposed by (Sacco, 2000). He proposed a model for dynamic taxonomies that is implemented by a series of menus where each menu item is associated with a restriction on the item base. The submenus on the following screen are dynamically determined and include only taxonomic concepts that are related to at least one remaining item in the information base. Thus opposed to the system of (Jannach, 2004) in (Sacco, 2000) each

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

navigation action of the user is annotated with exactly one restriction and empty query results are impossible due to dynamically determined navigation paths.

Furthermore, user expectations have to be taken into consideration when discussing different types of feedback elicitation and user interaction. There is a fundamental difference in users' attitudes between entering a community platform and accessing a commercial online shop. When interacting with a non-commercial collaborative filtering system, users know that they have to provide ratings in order to help build up community knowledge and to receive good recommendations themselves in return. So what they give and what they receive are both of non-monetary value. In a commercial environment they have to pay anyway, so there is no good reason for them to educate a recommender system in addition to making a purchase. Therefore, they expect to receive professional advice and good service as part of their shopping experience.

Trying to understand the users' needs and softly guiding them through a sales process is thus not only a form of preference elicitation, but also fulfills important functions with respect to sales psychology. The research on persuasiveness of recommender systems is still in its infancy (Fogg, 1999; Gretzel and Fesenmaier, 2006), but the basic idea is to design virtual sales assistants that intentionally influence online-shoppers' attitudes in order to sell products. As a first step, conversational systems engage in a question and answer style of dialogue or interactively propose items in order to reveal the preferences of their users (Linden et al., 1997; Smyth and Cotter, 2001; Shimazu, 2001; Burke, 2002; Torrens et al., 2002; Ricci et al., 2003; Jannach, 2004). Being explicitly asked about the importance of a specific feature of a product might influence users by itself. For instance, if users are forced to be explicit about the primary motives for their holiday trip (Ricci et al., 2003) or have to define the level of risk they are willing to take when deciding on a financial investment product (Felfernig and Kiener, 2005), might influence their viewpoints when they assess the utility value of an item that is recommended to them.

Having discussed research on these different aspects of recommender systems and the surrounding background literature, it is possible to compare the application of different recommendation strategies on user-models that solely consist of a limited set of explicit user requirements.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

3. Recommendation strategies for user requirements

(Adomavicius and Tuzhilin, 2005) formalized a recommendation problem by assuming an utility function *rec* that measures the usefulness of an item $i \in I$ to a user $u \in U$, i.e. $rec : U \times I \mapsto R$, where R is a totally ordered set of numbers within a certain range. Thus, the task of a recommender system is to identify for each user those items that follow an abstract goal such as maximizing the user's utility, increasing online conversion rates or even optimizing the user's lifetime value. Therefore, they define a recommendation task as finding the item i_j that maximizes the given user's utility.

As discussed in the previous section, different paradigms exist for the computation of this utility function. They differentiate themselves by the type of knowledge they exploit, i.e. community and product knowledge or some domain expert's heuristics. However, regardless of the underlying recommendation paradigm, all methods for personalized recommendation rely on a user model. As this work focuses in particular on personalizing short-length interactions with anonymous users, each user model is equivalent to a single user session. A user model is also not updated when the same physical customer reenters the website due to her anonymity. Furthermore, as motivated in the introduction, it is assumed that all algorithm variants have to work on the same user model, i.e. in this situation, solely a set of explicit user requirements constitutes the user model and is represented by attribute-value pairs. First, the following presents four different recommendation strategies: knowledge- and utility-based recommendation, user- and item-based collaborative filtering, as well as supervised association rule mining. As a second step, a case study is developed from usage data of a successfully fielded webshop and followed by a comparative evaluation of historic data. Content-based and demographic information filtering (Pazzani, 1999; Pazzani and Billsus, 2007) are not considered in this comparative review because they can't make personalized recommendations based on a set of explicit user requirements.

3.1. Knowledge-based recommendation

Figure 2 depicts a highly simplified example scenario. The knowledge-base consists of four personalization filters that are grouped into three different priority levels: low, medium and high. This signifies that not all applicable filters have to be satisfied when the system derives a recommendation, but rather that some priority-led relaxation takes place if necessary. Each personalization filter consists of a condition

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

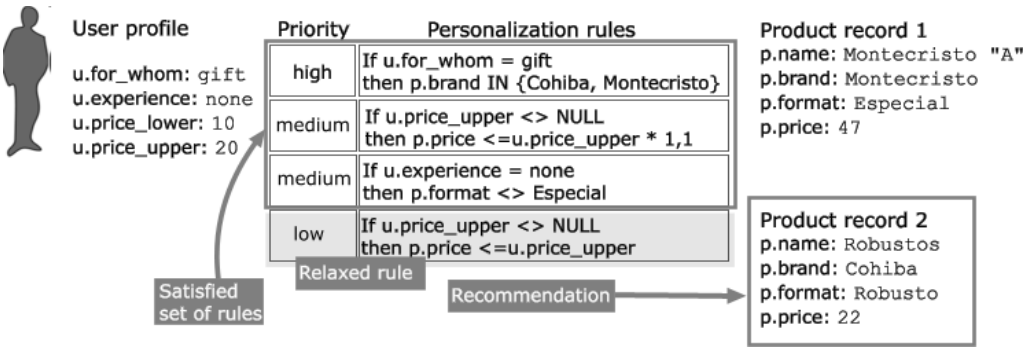

Figure 2. Example knowledge-based recommendation

(*If* ...), a consequent (*then* ...) and a priority value (e.g. *high*, *medium* and *low* in our example). Depending on the user model, the condition parts of a subset of all filters will evaluate to true and their consequents must be satisfied by all items in the result set. If none of the items in the product repository satisfies all applicable restrictions, then the lower priority personalization filters are stepwise relaxed until at least one item is part of the result.

Note, that in Figure 2 users as well as product characteristics are represented as attribute-value pairs in a pseudo object-oriented notation (u.attribute = value and p.attribute = value). Self explanatory abbreviations have been used such as *u.experience* signifying the user's smoking experience. The filter restrictions need to bridge the gap between abstract user requirements like looking for a gift or having no smoking experience and more technical product characteristics like prestige of brand names or different formats. Customers tend to rate personalization filters that derive from 'hard facts' such as 'a gift needs to have a prestigious brand name' as highly important, while 'soft' preferences like price expectations might be actually overruled once the customer understands why a specific product proposal clearly matches her needs. This is also the reason behind having two rules on the product price in our small example: the low-priority rule on the upper price limit may be among the first to be relaxed, but an additional rule at a medium priority level states that the customer's limit on the product price must not be exceeded by more than 10%. As a consequence the knowledge-based recommender proposes the second product record, i.e. the utility of item 2 for user u is higher than the utility of item 1 according to this knowledge-based recommender implementation: $rec_{kb}(u, 2) > rec_{kb}(u, 1)$. However, note that pure knowledge-based recommenders do not compute an utility score, but

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

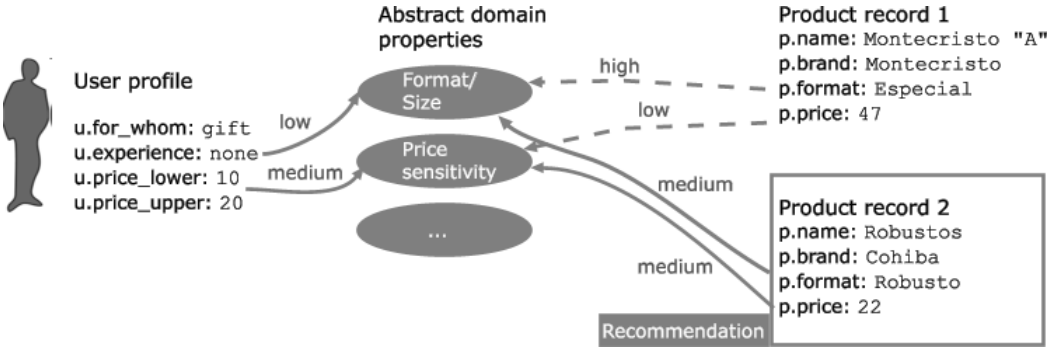

Figure 3. Example utility-based recommendation

decide if an item is part of their result set or not. Therefore, rec_{kb} is defined as follows:

$$rec_{kb}(i,u) = \begin{cases} 1 : kb \vdash i \\ 0 : \text{else} \end{cases} \quad (1)$$

The assumed utility of an item i for user u is 1 and therefore recommended if i is implied by the knowledge base and 0 otherwise.

3.2. Utility-based recommendation

Utility-based RS, like their knowledge-based counterparts, rely on domain knowledge to personalize the system behavior. However, they encode relationships between user characteristics and product features in an indirect way. Figure 3 depicts abstract domain properties or value dimensions such as *format* or *price sensitivity* that constitute the linking pin between customer requirements and product domain. For instance what is considered high-priced depends on domain-specific factors typically known by domain experts. When trying to measure the utility $rec_{ut}(i,u)$ a general evaluation scheme such as Multi-Attribute Utility Theory (MAUT) (von Winterfeldt and Edwards, 1986) can be adopted for estimating u's interest in item i:

$$rec_{ut}(i,u) = \sum_{p \in P} score_{u,p} \cdot score_{i,p}, \quad (2)$$

where P is the set of abstract domain properties, $score_{u,p}$ the estimated interest of u in a property $p \in P$ and $score_{i,p}$ the utility score an item i achieves with respect to p. $rec_{ut}(i,u)$ is then computed as the weighted sum of an item's property score with the estimated user interest in that property.

Thus, in the example in Figure 3 the second product record scores better with respect to modeled properties like *format* and *price sen-*

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

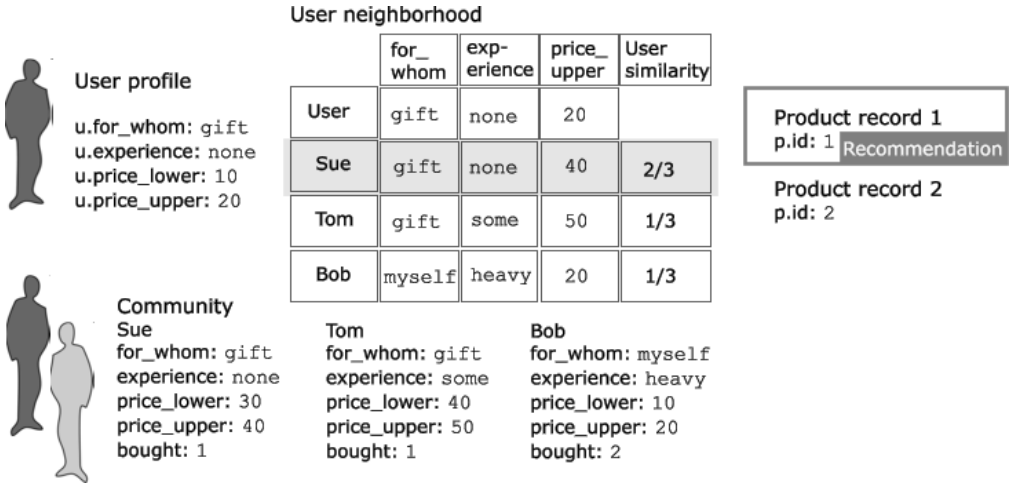

Figure 4. Example user-based collaborative filtering

sitivity: the user's score for *price sensitivity* is higher than for *format* (medium vs. low) and product 2 scores better with respect to *price sensitivity* than product 1.

3.3. User-based collaborative filtering

User-based collaborative filtering exploits similarities between users to make predictions. It neither requires the acquisition of domain knowledge prior to system deployment nor structured product descriptions. Therefore, in Figure 4 both product records are solely represented by a unique identifier. However, collaborative filtering assumes the existence of recorded community transactions. In a first step users similar to the current user are identified (i.e. user neighborhood formation). In nearly all collaborative recommender systems similarity between users is based on ratings. These can be either explicit ratings, for instance when users express how they liked a movie on a scale from 1 to 5, or implicit ratings when transactions such as buys or pageviews are extracted. In this case the attribute-value pairs of user profiles are interpreted as nominal ratings and user similarities are computed from the set of u's explicit requirements Rq_u. See (Basilico and Hofmann, 2004) for a unified approach that includes exploiting user characteristics for collaborative filtering.

The example as well as the experiments use Dice coefficient (Frakes and Baeza-Yates, 1992) to determine the similarity between two users u and v, i.e.:

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

$$sim(u,v) = \frac{2 \times |Rq_u \cap Rq_v|}{|Rq_u| + |Rq_v|} \tag{3}$$

It is computed as twice the number of similar requirements of u and v divided by the sum of requirements of both users. In Figure 4 user *Sue* has most requirements in common with the current user and therefore is her nearest neighbor (2 out of 3 requirements match). Thus, the product bought by *Sue* will also be proposed to the current user in a second step.

These experiments use the k-Nearest Neighbor method for deriving recommendations. Besides user attribute evaluations, the profile of each user v also contains a set of unary ratings on items R_v which were the user's choice (e.g. items that she actually added to her shopping basket). Thus $rec_{U2UCF}(u,i)$ computes most frequent successful transactions in u's neighborhood of users N_u, where the neighborhood size is limited by k, i.e. $|N_u| \leq k$.

$$rec_{U2UCF}(i,u) = \frac{\sum_{v \in N_u} score_{v,i}}{|N_u|}, \text{ where} \tag{4}$$

$$score_{v,i} = \begin{cases} sim(u,v) \,:\, i \in R_v \\ 0 : \text{else} \end{cases} \tag{5}$$

Note, that a variety of versions of collaborative filtering algorithms has been explored like presented in (Schafer et al., 2007). The approach taken here is memory-based user-to-user correlation, as the user neighborhood for a new user has to be computed during runtime of the system.

3.4. Item-based collaborative filtering

Item-based collaborative filtering was first proposed by Sarwar et al. (Sarwar et al., 2001). This technique precomputes the similarity between items based on how users rated them instead of using product characteristics. So high ratings from the same user for a pair of items is used as an indicator that these two items are similar to each other. However, we may only exploit explicitly formulated customer requirements in our experiments. Therefore, Figure 5 depicts a user/item matrix where the different answering options are also represented as items. User *Sue* for instance has positively "rated" the answers *gift* and *none* as well as purchased product number 1. The similarity between items i and j can thus be computed by determining the cosine of the angle between their rating vectors, i.e.:

$$sim(i,j) = cos(\overrightarrow{i}, \overrightarrow{j}) = \frac{\overrightarrow{i} \cdot \overrightarrow{j}}{|\overrightarrow{i}| \cdot |\overrightarrow{j}|} \tag{6}$$

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

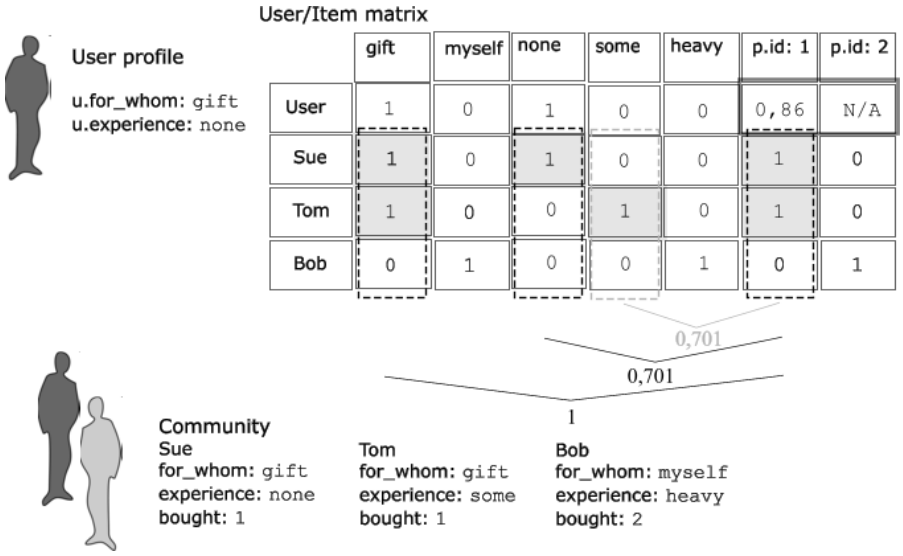

Figure 5. Example item-based collaborative filtering

Note, that in our experiments we only need to compute similarities between answering options and purchased items. Similarities between different products or between answering options are irrelevant for our purpose, as we only use the given answers of a user to determine those products she is likely to purchase.

The utility value of an item i for user u is computed as the sum of ratings in Rq_u (i.e. all explicit user requirements) weighted with their similarity to item i, i.e.:

$$rec_{I2ICF}(i,u) = \frac{\sum_{r \in Rq_u} sim(r,i) \cdot r}{|Rq_u|} \tag{7}$$

In Figure 5 the utility value of product 1 is 0.86 for the given user profile. The utility for product 2 is 0 as there are no users that selected the answering options 'gift' or 'none' and purchased product 2.

3.5. Association rules

Assume a set of items I and a set of transactions $Trans$, where each transaction $T_v \in Trans$ of user v is represented by a set of items $T_v \subseteq I$ that the user for instance bought during her online visit. An association rule has the form $X \Rightarrow A$, where X and A are sets of items from the same transaction $X, A \subseteq T_v$ with $X \cap A = \emptyset$. The *confidence* (c) of a rule signifies the conditional probability that a transaction containing X also contains A:

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

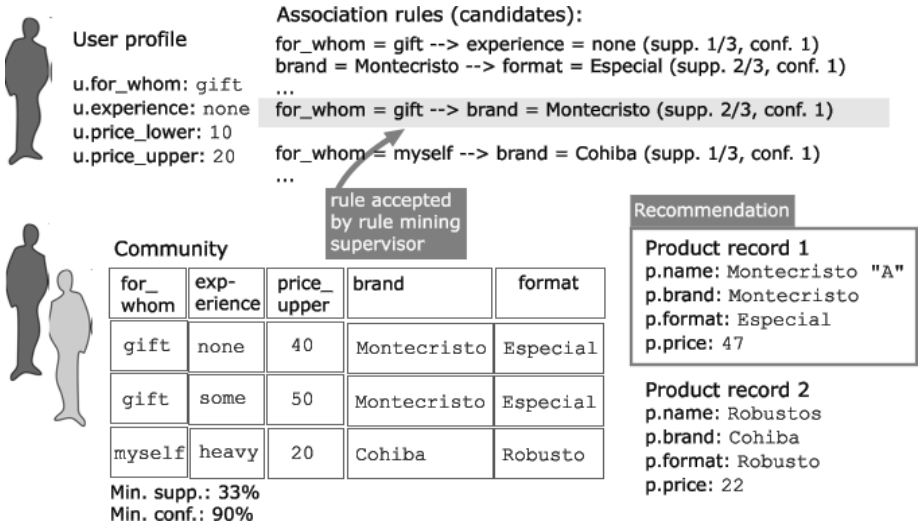

Figure 6. Example association rules

$$c(X \Rightarrow A) = \frac{|\{T_v|X, A \subseteq T_v\}|}{|\{T_v|X \subseteq T_v\}|} \tag{8}$$

I.e. the number of all transactions containing both, X and A, divided by the number of transactions containing X. While *confidence* measures the strength of a functional dependency, *support* (s) gives its relative frequency of occurrence:

$$s(X \Rightarrow A) = \frac{|\{T_v|X, A \subseteq T_v\}|}{|Trans|} \tag{9}$$

Figure 6 demonstrates the application of association rules to the community data. Note, that in this case the set of items I contains user requirements and features of products bought by the user. A comparable approach was taken by (Adomavicius and Tuzhilin, 2001) to build customer profiles. Each row in the *Community* table in Figure 6 represents a user transaction T_v. For instance the first row results from the user profile of *Sue*. She was looking for a gift and the recipient had no smoking experience. She finally selected the *Montecristo "A"* model that is big size. The goal is then to find functional dependencies between these explicit user requirements and the features of products that actually interested the user during her online visit. Thus, instead of building a recommendation knowledge base from heuristics and expert knowledge, these rules could be learnt from past transactions. Although the rule mining process can be guided by setting lower limits on *support* and *confidence* measures, an unsupervised learning approach is not

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

recommended. First, a lot of dependencies between user requirements or between product features might be found that are useless for the specific goal of mining recommendation rules (e.g. *brand* = *Montecristo* ⇒ *format* = *Especial*). Therefore, contrasting (Sarwar et al., 2000), a domain expert will have to check the association rules for plausibility and validity. These issues were also observed by (Adomavicius and Tuzhilin, 2001) when trying to discover a set of rules that describe customers' behavior. In the very simplified example scenario depicted in Figure 6 only a single rule ("The well known brand *Montecristo* is often bought for the purpose of a gift") is identified and applied to the product database. Thus product record number 2 would thus be recommended.

3.6. Hybrid algorithm variants

Hybrid recommendation algorithms are often employed in order to overcome shortcomings or improve performance of one of the base types of recommendation strategies. Burke distinguished seven different forms of hybridization (Burke, 2002). However, this work concentrates only on three variants: *weighted*, *cascade* and *switched*. For comparing the other hybridization techniques, too, either more information-rich user models would be necessary for experimentation or specific implementation decisions would need to be made.

The weighted method combines the scores of several recommendation techniques together and produces a ranked list of recommendations on its own:

$$rec_{weighted}(i,u) = \frac{\sum_{t_j \in Rec} w_{t_j} \cdot rec_{t_j}(i,u)}{\sum_{t_j \in Rec} w_{t_j}} \tag{10}$$

For each t_j out of a set of recommendation techniques Rec the utility of recommending item i to user u is derived by $rec_{t_j}(i,u)$. The overall utility $rec_{weighted}(i,u)$ is computed as a weighted and normalized sum, where w_{t_j} is the weighting factor of technique t_j. Although the implementation of a weighted hybrid is quite straightforward, it follows the implicit assumption that all techniques contribute uniformly to the overall result across the item space (Burke, 2002). Cascade hybrids avoid this effect by implementing a staged process. They sequentially order the techniques, where each succeeding recommender only refines the output of its predecessor.

Assume a sequence of techniques $Rec = \langle t_1, \ldots, t_k \rangle$, where t_1 is the recommender with highest and t_k with lowest priority, then:

$$rec_{cascade}(i,u) = rec_{t_k}(i,u) \tag{11}$$

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

and $\forall t_j$, with $j = 2 \ldots k$ must hold:

$$rec_{t_j}(i,u) = \begin{cases} rec_{t_j}(i,u) \ : \ rec_{t_{j-1}}(i,u) \neq 0 \\ 0 : \text{else} \end{cases} \tag{12}$$

Thus in a cascade hybrid all involved techniques except the first one can only change the ordering of the list of recommended items from their predecessor or exclude an item by setting its utility to 0. However, they may not introduce items to the recommendation list that have already been excluded by one of the higher priority techniques.

As knowledge-based recommenders produce unsorted recommendation lists cascading them with another technique for sorting results is popular. Examples of *EntreeC*, a knowledge-based restaurant recommender that is cascaded with a collaborative filtering algorithm (Burke, 2002) or the knowledge-based Advisor Suite recommender shell that includes a utility-based sorting scheme (Jannach, 2004) come to mind.

In addition we introduce a $method_1$-*relax-method*$_2$ N switching hybridization strategy (short: $switch_{m_1 r x m_2 N}$) to address situations where one method does not produce enough recommendations for each user. For instance, cascading strategies do have the unfavorable property of potentially reducing the recommendation set with each additional technique applied. As will be shown in the experiments, situations can arise where cascade algorithms do not deliver the maximum number of propositions and therefore deteriorate accuracy. $switch_{m_1 r x m_2 N}$ addresses this issue by relaxing $method_1$ once the recommendation set size falls below N and steps back to a $method_2$.

$$rec_{switch_{m_1 r x m_2 N}}(i,u) = \begin{cases} rec_{m_1}(i,u) \ : \\ \quad |\{j | j \in I \wedge rec_{m_1}(j,u) > 0\}| \geq N \\ rec_{m_2}(i,u) \ : \text{else} \end{cases} \tag{13}$$

The $switch_{m_1 r x m_2 N}$ variant works like method m_1, if the threshold N on the recommendation set is reached. Otherwise it switches to method m_2.

Having discussed how different recommendation strategies might exploit explicit user requirements for generating personalized product propositions, the next section provides a comparative evaluation on actual user transaction data.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

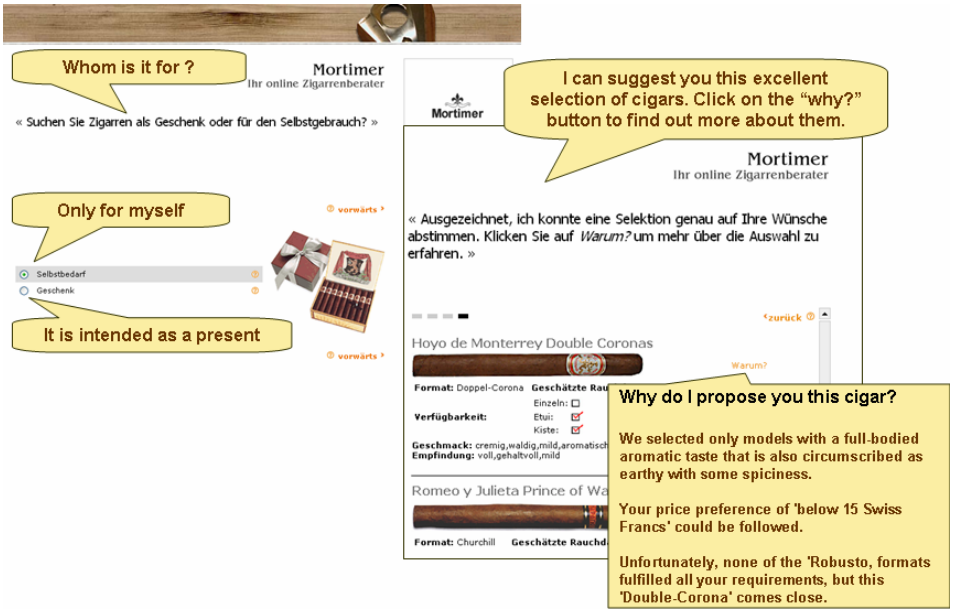

Figure 7. Screenshots of cigar advisor

4. Case study: premium-cigars.ch (PC)

Premium-cigars.ch is a Swiss online store offering an assortment of more than 140 types of fine cigars. Courtesy of the shop owner, the authors were allowed to perform experiments on anonymous transaction data and develop improvements for the deployed system.
The following research questions are posed:

1. What characteristics do different recommendation strategies possess with respect to accuracy and catalog coverage when only user feedback in the form of explicit requirements is available?

2. What improvements can be achieved by combining methods with *weighted*, *cascade* and $switch_{m_1 r x m_2 N}$ hybridization strategies?

4.1. Background

A virtual sales assistant termed *Mortimer* was fielded on this platform in 2003[2]. Figure 7 depicts how the system converses with its users by asking them about their requirements and preferences. Questions like: *"who is it for?"*, *"how experienced are you in smoking cigars?"* and *"what tastes and effects do you prefer?"* help the system to estimate

[2] Visit the system at *http://www.premium-cigars.ch*.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

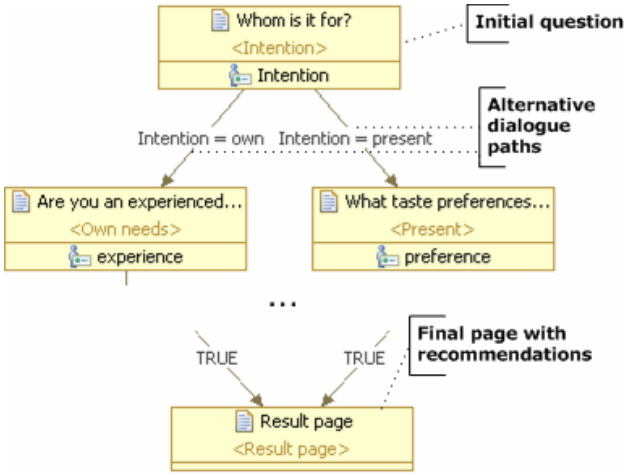

Figure 8. Exemplary process flow

the relative importance of concepts like prestige, exclusivity or price to the user and enable the knowledge-based recommender to recommend cigars that promise an enjoyable taste and a smoking experience without regret.

With the help of domain expertise a total of 47 filter rules were initially defined that can be grouped into the following topics (number of rules in brackets):

- Inexperienced smokers (8): these rules ensure that cigars which are not advisable to beginners due to their straining effects or long smoking duration will be excluded from recommendations to inexperienced smokers.

- Cigar format (7): 16 different formats of cigars like *Corona* or *Churchill*[3] exist. These rules ensure that users' requirements regarding size and smoking duration are followed; similar formats may be substituted if necessary.

- Taste (11): the system is aware of 27 different tastes and flavors a smoker could experience when consuming a cigar.

- Effect (12): this category describes the impact a cigar has on the smoker. 23 different expressions denote these effects.

- Unit (2): each type of cigar can be offered as a single unit or in boxes with 5 or 25 pieces.

[3] Named after the famous British prime minister.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

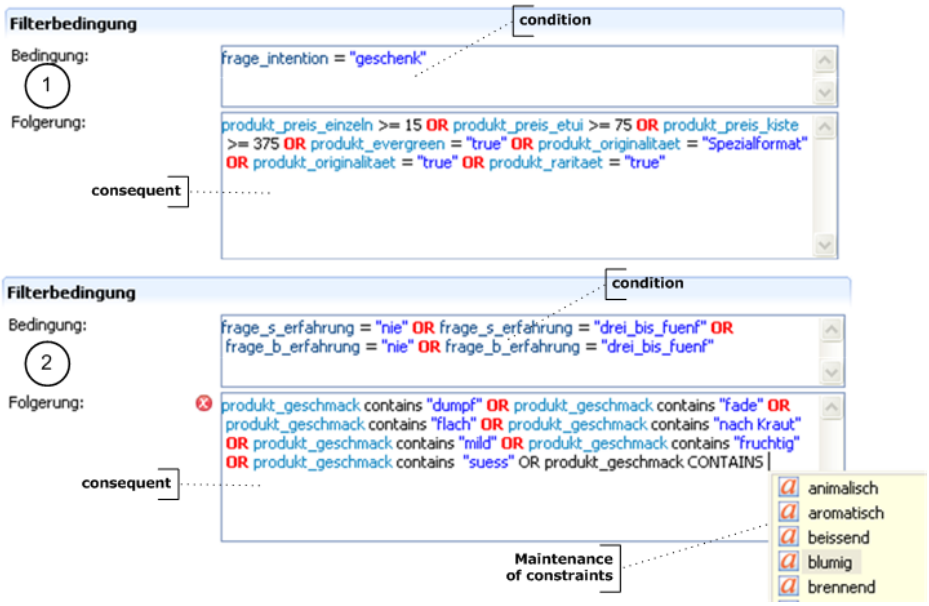

Figure 9. Screenshots of constraint editing environment

– Price (7): pricing rules observe user's requirement on the price. For instance if the user selects a range from $10 to $15 per piece but none of the cigars within that price range satisfies all other criteria, then prices slightly below or above that range will also be acceptable for the system.

The knowledge base was developed in 2003 and is since then in use to give knowledgeable recommendations to online users. Figure 8 gives an exemplary pageflow definition of the elicitation dialogue. It is enacted by the conversational *Advisor Suite* recommender (Jannach, 2004). With respect to filter constraints the editor environment is depicted in Figure 9. E.g. on the upper half (1) the following filter condition is depicted: "if the user requires a gift (condition part) then the price for a single type of cigar should be above $15 or for a packaging unit with 5 pieces above $75 or for a box of 25 pieces above $375 or an *Evergreen* type of cigar or ... (consequent part)". On the lower half (2) a filter restriction that selects appropriate types of cigars for inexperienced users is defined. It filters hot and strong tastes out.

This short outline of the knowledge base should give an idea of its functioning and make clear that the problem is too complex to be addressed by parametric search forms. Besides making recommendations the system also educates its users by explaining them the reasons why an item is proposed, i.e. which heuristics or rules have been applied.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

A previous evaluation already investigated the question if the advice given by the virtual shop assistant affects the buying behavior of online shoppers (Zanker et al., 2006). Here, it was possible to compare the sales records in the period before the introduction of the conversational recommender system with those in the period afterwards. One interesting finding was that the lists of the top ten ranked products in the two periods differed considerably. Some types of cigar that were infrequently sold in the period before the introduction of the recommender system experienced a very high demand afterwards. When digging deeper it turned out that these types had also been recommended very often, thus the increase in demand for some products is positively correlated with how often the system recommends them. Therefore it must be noted that a minor bias in favor of the knowledge-based recommender will be present in the data. As real user interactions were recorded using the conversational recommender, these users also received knowledge-based recommendations. Consequently, these recommended items will have a higher chance of being put into the shopping basket during that session resulting in bias.

Nevertheless this evaluation focuses on comparing the accuracy of different recommendation strategies using only the explicit requirements collected by the sales advisor as the primary form of user feedback. The aim is to explore different (hybrid) algorithm variants in order to gain more insight into their strengths and weaknesses in commercial environments as well as improve the recommendation accuracy of future versions of the deployed RS.

4.2. Experimental setup

The evaluation is based on a dataset extracted from weblog information and profiling data of the conversational recommender system over a period of 16 months (Oct/05 - Jan/07). More than 30 thousand distinct user sessions were identified. In 548 sessions users actually interacted with the conversational recommender system until they received a recommendation. Although more than 548 accesses to the conversational recommender were recorded, many were discarded as they could not be unambiguously related to entries in the weblog. In addition to answers to the RS, the research required the set of products that users were actually interested in, which had to be obtained from the weblog. Adding an item to the shopping basket or accessing the detailed description of a product for more than half a minute was used as an indicator for user interest. For this reason, data extraction from weblog files and the application of several preprocessing steps like identifying unique user sessions or removing entries deriving from web crawlers was necessary.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

However, we omit a further description of the applied techniques as a comprehensive discussion on data extraction and preprocessing from weblogs was done by (Mobasher et al., 2000; Mobasher, 2007).

Furthermore all user sessions that showed interest in more than 12 products were eliminated, as the relationship between explicit requirements and observed interest in items would become too noisy. So finally a dataset comprising 189 different user sessions (U) with a total of 1515 explicit user requirements and 631 observed item transactions (set of item ratings R for deriving predictions) on a total of 143 different items (I) remained. The sparsity of the dataset was computed as the share of empty entries in the user-item matrix $1 - \frac{|R|}{|I| \cdot |U|}$ which resulted in a sparsity level of 97,67% ($|R| = 631$, $|I| = 143$ and $|U| = 189$). Note, that the number of explicit user requirements does not impact the sparsity level itself as only product items will have to be predicted.

The first step compared the following six recommendation strategies.

- Unpersonalized 'Top n' recommender based on actual sales records (Top_n)
- Utility-based recommender (Ut)
- Knowledge-based sorted by price in ascending order (KB_{price})
- Knowledge-based sorted by utilities (KB_{Ut})
- User-based collaborative filtering ($U2UCF$)
- Item-based collaborative filtering ($I2ICF$)
- Association rules with supervision sorted by price in ascending order (AR)

Top_n is the baseline algorithm, computing an unpersonalized ranked list of product propositions from top selling items. The exact ranking derives from sales records during that period.

Ut exploits utility definitions on a MAUT scheme (von Winterfeldt and Edwards, 1986) that has been defined by a domain expert. It ranks models from more well known producers like *Cohiba* or *Montecristo* higher for users who are looking for a gift. These brands are deemed to be more prestigious in the eyes of the recipient. In addition, it considers price and unit preferences of the user. So, if the user formulated these requirements then cheaper types of cigars or those available in smaller packaging units are higher ranked.

KB_{price} and KB_{Ut} are cascade hybrids that reason on the knowledge base outlined in Subsection 4 to classify items as recommendable or not.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Both knowledge-based algorithm variants relax their filter conditions if the result set would be empty otherwise. They relax low priority rules first, such as preferences on cost and packaging units. KB_{price} sorts results by ascending prices and KB_{Ut} ranks them according to the personalized utility scheme of Ut.

$U2UCF$ is a collaborative filtering method that exploits user-to-user correlations on explicit customer requirements as presented in Subsection 3.4. Recommendations are derived by identifying the most frequent items of the k nearest neighbors (see equations (4) and (5) for reference). k was set to 30 after conducting a sensitivity analysis.

AR represents a knowledge-based recommender that builds on functional dependencies that derive from an association rule mining attempt as discussed in subsection 3.5. Although most retrieved rules were deceptive and therefore discarded, five interesting patterns have been found that form the basis for this algorithm variant:

- Users with medium price sensitivity ($10 to $20 per piece), order *Cohiba* brands.
- Users asking for proposals that are available in units of 5 pieces also buy units of 5 pieces.
- Users asking for types of cigars with an *intense* taste, actually also buy these types of cigars.
- If users are looking for a gift and they do not know how often the recipient smokes cigars, they purchase those types of cigars that are available in units of 5 pieces.
- Furthermore, these users give *Cohiba* brands as presents.

It is quite interesting to see that some of the domain expert's rules in the knowledge base were validated by the association rule mining exercise. For instance it is no surprise to discover that obviously experienced users looking for intense tastes will stay with these types of cigars or users in need for a gift for someone with unknown (and factual few) smoking experience show interest in the popular *Cohiba* brand or in cigars in smaller units. As results will show in the next subsection, these few functional dependencies with high confidence (above 90%) performed relatively well. For the rule mining exercise itself we employed the Weka workbench (Witten and Frank, 2005).

A second exercise step applied the three hybridization strategies *weighted*, *cascade* and $switch_{m_1 x m_2 N}$ on the most promising algorithm combinations and explored their impact on the accuracy and diversity of the recommendation lists.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

4.3. Methodology

The set of items that caught users attention, represented as unary ratings (R), were used to evaluate the different algorithm variants. For better understanding the following table gives the user model of an arbitrary session:

User model (id: 262.691)	
Intent:	*for myself*
Smoking experience:	*monthly*
Taste:	*mild*
Effect:	*encouraging*
Size:	*Robusto*
Unit:	*don't care*
Price:	*don't care*
Codes of rated items (R):	{60, 265}

As all algorithms are only allowed to use explicit user requirements Rq for learning, the *Given 0* evaluation method, that all rated items R are used for testing and none of them for learning, was employed, i.e.:

$$testset_u = R_u$$
$$learnset_u = Rq_u$$

Using all of u's explicit requirements Rq_u for learning and testing on the full rating set R_u does not necessitate a cross validation with several experiment runs because no random selection of the test or learning set is involved. For each user at most ten recommendations are computed, forming an ordered sequence of recommendations, where for all recommended items no other item with a lower index and a higher recommendation score - according to technique t - must exist:

$$recset_u = \langle i_1, \ldots, i_n \rangle, \forall i_k, i_l \in I : k < l \rightarrow rec_t(i_k, u) > rec_t(i_l, u) \tag{14}$$

The size of the recommendation set n was 10 for all hybrid experiments; the baseline algorithms have been evaluated with $n = 3$, 5 and 10. It is assumed that recommendations contained in the testset are successful hits, i.e. $hits_u = recset_u \cap testset_u$. The accuracy of recommendations

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

is computed using the Precision (Prec), Recall (Rec) and F1 metrics (Sarwar et al., 2001; Herlocker et al., 2004):

$$Prec = \frac{|hits_u|}{|recset_u|}$$
$$Rec = \frac{|hits_u|}{|testset_u|}$$
$$F1 = \frac{2 \cdot Prec \cdot Rec}{Prec + Rec}$$

The Precision metric gives the share of successful recommendations from the total number of computed recommendations, while the Recall metric computes the ratio of hits and the theoretical maximum number of hits due to the testing set size. The average test set size is 3.3 ($\frac{|R|}{|U|}$). Consequently, if it were possible to predict all items in the testing set, Recall would be 100% but Precision only around 33%! The F1 measure combines both Precision and Recall. Another important metric in our evaluation is the catalog coverage ($Ccov$):

$$Ccov = \frac{|\bigcup_{u \in U} recset_u|}{|I|} \tag{15}$$

It is represented by the share of items that are ever recommended to a user (Herlocker et al., 2004) and can be used for analyzing the diversity of recommendations. Nevertheless, improving accuracy remains the primary interest.

4.4. Results

The first step compared the six recommendation algorithms outlined in Subsection 4.2 in terms of accuracy and catalog coverage (see Table I). All experiments have been conducted with $n = 3$, 5 and 10. It can be observed that when the size of the recommendation set n increases, the Recall also increases but Precision drops. This behavior indicates that the higher ranked recommendations are more probable to become a successful hit than the lower ranked ones and therefore the algorithm and its implementation work well. To the contrary, Precision of KB_{price}, $I2ICF$ and AR augments with increasing n. Interestingly, KB_{price} and AR do not have personalized ranking schemes, but filter conditions that state if an item is within the result set or not (compare Subsection 3.1). Therefore, these two variants need a sort criteria and we selected an ascending order on *price* for this purpose.

Top_n was the baseline algorithm and recommended the n most frequently sold items to all users in an unpersonalized way. The relative

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Table I. Results on base algorithms

Strategy	n	$Prec$	Rec	$F1$	$Ccov$
Top_n	3	10.05%	9.03%	9.51%	2.09%
Top_n	5	7.62%	11.41%	9.13%	3.49%
Top_n	10	5.24%	**15.69%**	7.85%	6.99%
Ut	3	4.23%	3.80%	4.00%	6.99%
Ut	5	3.70%	5.55%	4.44%	11.19%
Ut	10	5.82%	17.43%	8.72%	17.48%
KB_{price}	3	8.60%	6.25%	7.24%	65.03%
KB_{price}	5	8.77%	8.65%	8.71%	77.62%
KB_{price}	10	10.04%	14.90%	11.51%	**89.51%**
KB_{Ut}	3	14.81%	**11.57%**	12.99%	65.73%
KB_{Ut}	5	12.97%	14.74%	13.80%	76.22%
KB_{Ut}	10	11.44%	**20.51%**	14.66%	**87.41%**
$U2UCF$	3	12.17%	**10.94%**	11.52%	27.27%
$U2UCF$	5	9.95%	14.90%	11.93%	36.36%
$U2UCF$	10	8.47%	**25.36%**	12.70%	**51.05%**
$I2ICF$	3	2.12%	1.90%	2.00%	44.06%
$I2ICF$	5	3.07%	4.60%	3.68%	56.64%
$I2ICF$	10	2.96%	8.87%	4.44%	77.62%
AR	3	2.65%	2.38%	2.51%	4.20%
AR	5	2.33%	3.49%	2.79%	6.99%
AR	10	2.43%	7.29%	3.64%	13.99%

small size of 143 products produced a Recall close to 16% for $n = 10$. Not surprisingly Catalog coverage remained below 7%, as each user was presented with the same set of top-selling items. The KB_{price}, $I2ICF$ and AR algorithms fall below the baseline in terms of accuracy. Again, the unpersonalized ranking scheme of KB_{price} and AR might be a reason for the bad results. In addition, AR works only on a small set of 5 rules that are not very discriminating. $I2ICF$ on explicit user requirements was lagging behind expectations and did significantly worse than its user-based counterpart. Computing similarities between answering options and actually purchased items ($I2ICF$) seemed to be a less efficient approach than recommending items other users with rather similar sets of explicit requirements have bought. A hybrid ap-

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Table II. Results on *weighted* hybrids

Strategy		*Prec*	*Rec*	*F*1	*Ccov*
AR/KB_{Ut}	*weighted*	6.19%	18.54%	9.28%	75.52%
$AR/U2UCF$	*weighted*	6.14%	18.38%	9.20%	53.14%
$AR/I2ICF$	*weighted*	3.02%	9.03%	4.53%	65.73%
$KB_{Ut}/U2UCF$	*weighted*	**9.05%**	**27.10%**	**13.57%**	72.73%
$KB_{Ut}/I2ICF$	*weighted*	5.89%	17.43%	8.80%	68.53%
$U2UCF/I2ICF$	*weighted*	5.77%	17.27%	8.65%	84.62%

proach between $I2ICF$ and AR remains for future work, i.e. computing similarities between answering options and features of purchased items.

The knowledge and utility-based hybrid (KB_{Ut}) is the system actually deployed on the platform. It outperforms the baseline algorithm by nearly 5% in terms of Recall. However, it is quite sobering that the high knowledge acquisition effort for the setup of the system did not pay off when compared to collaborative filtering that clearly reaches better results in terms of accuracy. Looking at the top 3 recommendations KB_{Ut} and $U2UCF$ achieve equal results, but with respect to top 10 $U2UCF$ performs better. Only the diversity of recommendation sets is weak for $U2UCF$, as due to the relatively small size of item ratings in user neighborhoods, derived recommendations do not vary much. In contrast, the knowledge-based algorithm potentially proposes nearly every item.

Note, that no figures are given for user coverage. It is defined as the relative share of users from the dataset that received recommendations. As user coverage was above 99% in all experiments, we omit it in the tables of results.

During the second step we ran all *weighted* and *cascade* experiments for the 4 base algorithms: AR, KB_{Ut}, $U2UCF$ and $I2ICF$. Table II gives results for the *weighted* variants, where intermediate recommendations from each of the two hybridized algorithms have been equally (1.0:1.0) weighted. Interestingly, all hybrids do worse in terms of accuracy than the more accurate algorithm of the two combined alone, except for the $KB_{Ut}/U2UCF$ that performs best. Table III outlines the results of a sensitivity analysis of weighting the recommendations of KB_{Ut} and $U2UCF$. In this case a ratio of 0,9 to 1,0 promises even higher accuracy and a reasonably good catalog coverage. It is quite interesting to see how the latter continuously improves with the relative weight of the knowledge-based strategy from 50,53% to 91,63%.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Table III. Results of sensitivity analysis on $KB_{Ut}/U2UCF$ *weighted* hybrids

Strategy		*Prec*	*Rec*	*F*1	*Ccov*
$KB_{Ut}/U2UCF$	*weighted 0.1/1.0*	8.84%	26.47%	13.25%	**50.53%**
$KB_{Ut}/U2UCF$	*weighted 0.2/1.0*	8.99%	26.94%	13.48%	51.05%
$KB_{Ut}/U2UCF$	*weighted 0.3/1.0*	8.99%	26.94%	13.48%	53.15%
$KB_{Ut}/U2UCF$	*weighted 0.4/1.0*	9.05%	27.10%	13.57%	55.94%
$KB_{Ut}/U2UCF$	*weighted 0.5/1.0*	8.89%	26.62%	13.33%	60.84%
$KB_{Ut}/U2UCF$	*weighted 0.6/1.0*	8.84%	26.47%	13.25%	64.34%
$KB_{Ut}/U2UCF$	*weighted 0.7/1.0*	8.94%	26.78%	13.40%	65.73%
$KB_{Ut}/U2UCF$	*weighted 0.8/1.0*	9.05%	27.10%	13.57%	67.83%
$KB_{Ut}/U2UCF$	*weighted 0.9/1.0*	**9.21%**	**27.58%**	**13.81%**	69.93%
$KB_{Ut}/U2UCF$	*weighted 1.0/1.0*	9.05%	27.10%	13.57%	72.73%
$KB_{Ut}/U2UCF$	*weighted 1.0/0.9*	8.68%	25.99%	13.01%	77.62%
$KB_{Ut}/U2UCF$	*weighted 1.0/0.8*	8.52%	25.52%	12.77%	80.42%
$KB_{Ut}/U2UCF$	*weighted 1.0/0.7*	8.47%	25.36%	12.70%	84.62%
$KB_{Ut}/U2UCF$	*weighted 1.0/0.6*	8.41%	25.20%	12.61%	86.01%
$KB_{Ut}/U2UCF$	*weighted 1.0/0.5*	8.25%	24.72%	12.37%	88.11%
$KB_{Ut}/U2UCF$	*weighted 1.0/0.4*	8.52%	25.52%	12.77%	90.21%
$KB_{Ut}/U2UCF$	*weighted 1.0/0.3*	8.62%	25.83%	12.93%	90.91%
$KB_{Ut}/U2UCF$	*weighted 1.0/0.2*	8.57%	25.67%	12.85%	90.91%
$KB_{Ut}/U2UCF$	*weighted 1.0/0.1*	8.41%	25.20%	12.61%	**91.63%**

Comparing the *cascade* hybrids was the next step and results are given in Table IV. It was initially thought that using a knowledge-based recommender and having its result set sorted by collaborative filtering in a cascade style would lead to a new recommender with much better performance. Astonishingly, accuracy deteriorated and Recall dropped below 19%, while Precision reached an ever high of nearly 14%, which means that two fifths of the theoretically maximal Precision being 33% have already been reached. The reason for this effect lies in the dramatic drop of the average size of recommendation sets.

While all other recommendation strategies could produce recommendation sets with exactly 10 items, the $KB_{Ut}/U2UCF$ *cascade* experiment design delivered on average only 4,86 recommendations. These were obviously more precise, but failed on Recall. As already discussed in Subsection 3.6 the *cascade* approach has the property of narrowing down recommendation sets. The knowledge-based recommender with 47 rules is already quite discriminating and collaborative filtering eliminates in a further step those items that cannot be derived from a user's neighborhood.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Table IV. Results on *cascade* hybrids

Strategy		*Prec*	*Rec*	*F*1	*Ccov*
AR/KB_{Ut}	*cascade*	11.67%	19.87%	14.70%	86.71%
$AR/U2UCF$	*cascade*	**8.62%**	**25.83%**	**12.93%**	51.04%
$AR/I2ICF$	*cascade*	3.17%	9.51%	4.76%	77.62%
KB_{Ut}/AR	*cascade*	10.08%	15.02%	12.06%	90.20%
$KB_{Ut}/U2UCF$	*cascade*	**13.93%**	**18.92%**	**16.05%**	72.03%
$KB_{Ut}/I2ICF$	*cascade*	12.51%	12.78%	12.64%	71.32%
$U2UCF/AR$	*cascade*	**3.65%**	**10.94%**	**5.47%**	36.36%
$U2UCF/KB_{Ut}$	*cascade*	**13.49%**	**19.20%**	**15.85%**	72.72%
$U2UCF/I2ICF$	*cascade*	3.97%	11.89%	5.95%	82.52%
$I2ICF/AR$	*cascade*	3.02%	9.03%	4.53%	58.04%
$I2ICF/KB_{Ut}$	*cascade*	13.02%	14.36%	13.66%	74.83%
$I2ICF/U2UCF$	*cascade*	6.77%	20.29%	10.15%	79.72%

Table V. Results on $switch_{crxw10}$ hybrid

Strategy		*Prec*	*Rec*	*F*1	*Ccov*
$KB_{Ut}/U2UCF$	$switch_{crxw10}$ *0.9/1.0*	**9.74%**	**29.16%**	13.90%	74.13%

$AR/U2UCF$ reached best accuracy results (25,83%), although they are only slightly better than $U2UCF$ alone. Further note, that the $KB_{Ut}/U2UCF$ as well as the $U2UCF/KB_{Ut}$ reached similar high values on Precision, but $AR/U2UCF$ and $U2UCF/AR$ did not show symmetric behavior. This lets us conclude that in the latter case the unpersonalized ranking scheme of AR massively deteriorates results.

Consequently, we can outline that the $KB_{Ut}/U2UCF$ *weighted* strategy considerably improves Recall, while the $KB_{Ut}/U2UCF$ *cascade* strategy produces very precise results, but has the disadvantage of narrowing down results too severely for some users. A strategy to circumvent the narrowing down property of a *cascade* hybrid without loosing too much of its precise predictions is a switching strategy between both. Therefore we propose a switching strategy $switch_{crxwN}$ that applies *cascade* ($m_1 = c$) if the number of recommendations does not fall below a threshold N and switches to *weighted* ($m_2 = w$) if it does. Table V indicates that this strategy, configured with a threshold of 10, doesn't only perform best with respect to accuracy but also

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

in terms of Catalog coverage when compared to all other hybrids. As Precision is close to 10% and 10 recommendations are given, we can say that on average the algorithm predicts one successful hit for each user.

To summarize, the case study on premium-cigars.ch delivered evidence for the following hypotheses:

1. Explicit customer requirements can be used to develop predictive models for personalized product recommendations where KB_{Ut} and $U2UCF$ proved to be the most promising techniques.

2. Standard user-to-user collaborative filtering outperforms knowledge-based approaches in terms of Recall once a dataset is large enough for building predictive models.

3. Knowledge-based techniques produce very precise recommendation lists, if they do not need to relax too many constraints. Cascading them with $U2UCF$ even improves on Precision.

4. Weighting KB_{Ut} and $U2UCF$ techniques leads to an overall improvement in terms of Recall.

5. A switching strategy between two techniques combines favorable characteristics from both.

In the following section it is explored how far these findings are generalizable or observable in different domains.

5. Additional evaluation studies

The validity of the findings presented above was tested using the data collected between November 2006 and November 2007 from a German online store[4] for consumer electronics and computer hardware. The store offers its customers standard catalog-based shop navigation for exploring the product assortments. In addition, it employs a knowledge-based recommender systems based on the Advisor Suite system (Jannach, 2004) to provide advice to users interested in notebooks (NB) or digital cameras (DC). In comparison to the cigar case-study, only users who interacted with one of the conversational sales advisors and showed considerable interest in at least one product item were included in the appropriate dataset. An *add to basket* or an explicit *request for availability* action was interpreted as an implicit positive

[4] Visit the system at *http://www.bitsuperstore.de.*

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

rating for a product item. Therefore, the DC and NB datasets contain only user sessions with at least one rated item and a set of explicit requirements from the respective domain. The following table compares the characteristics of the two additional datasets with the cigar dataset PC:

Dataset	$\|U\|$	$\|I\|$	$\|R\|$	$\|R_q\|$	Sparsity level
PC	189	143	631	1515	97,67%
DC	75	83	124	859	98,01%
NB	206	272	363	3410	99,35%

Although the tobacco domain is unrelated to consumer electronics or computers the datasets show similar characteristics with respect to the size of the product base ($|I|$) or the sparsity level of the rating matrices. However, when comparing the average number of explicit requirements per user ($\frac{|R_q|}{|U|}$) we can observe that they increase with the domain complexity, i.e. around 8 in the PC domain, 11 for DC and above 16 for the notebooks. Not surprisingly, a conversational recommender for notebooks may ask users for more preferences than a system proposing cigars.

Both recommenders offer two different interactions styles: The first one elicitates abstract domain requirements such as the proposed use of the product item (e.g. gaming or office work in the notebook domain), mobility requirements or a general preference for low-cost offers. The second interaction style requires the user to specify technical product parameters like *memory* and *display size* for notebooks or *resolution* and *optical zoom* for digital cameras.

Again, the knowledge-based recommender represents the recommendation technique that is actually deployed in the webshop. The digital camera advisor contains 45 personalization filters that encode for instance the following domain knowledge:

- Determine the appropriate *resolution* based on the intended use of the pictures taken. For instance, if the user wants to print them as posters the system proposes products with higher pixels counts.

- Propose cameras that run on standard AA-size batteries, if a high operational readiness is important.

- Ensure that parametric search criteria like *manufacturer*, *size dimensions* or *digital zoom* are fulfilled.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Table VI. Results on base algorithms, $n = 10$

Dataset	Strategy	*Prec*	*Rec*	*F*1	*Ccov*
DC	*U2UCF*	3.47%	**21.48%**	5.97%	51.81%
DC	KB_{Ut}	**12.28%**	17.86%	14.55%	40.96%
NB	*U2UCF*	2.14%	**13.43%**	3.69%	74.63%
NB	KB_{Ut}	2.18%	10.92%	3.63%	37.87%

In comparison, the notebook advisor includes 50 filter constraints that encode domain expertise such as:

- Propose one GB of main memory if the customer wants to run computer games or graphical applications.
- If the notebook will be primarily used for travel, low weight and long lasting batteries are important.
- Select products that conform to the selected parametric search criteria such as *processor type*, *size of LCD panel* or *optical drives*.

Analogously to the cigar case study, the KB_{Ut} algorithm matches the explicit requirements of the user model with the condition part of the personalization filters, which if fulfilled restricts the retrieved products accordingly. Again, if a single item does not match all the applicable restrictions the filter restrictions are relaxed according to a priority scheme. The recommendation list is finally sorted by a personalized item utility function that takes user preferences into account. The *U2UCF* algorithm and all hybrid variants exploit the same user models as KB_{Ut} (as described in Section 3).

Here, we report only on the experiments involving the *U2UCF* and KB_{Ut} algorithms as these methods perform best in the *PC* study. Algorithm parameters, reported accuracy metrics and hybridization variants were kept constant to ensure comparability, e.g. maximal size of recommendation set n equalled to 10 and neighborhood size for *U2UCF* was set to 30. Table VI summarizes the results of the base algorithms for both datasets. In the digital camera domain *U2UCF* produces higher Recall and KB_{Ut} performs better in Precision. The reason becomes evident when looking at the average size of the recommendation set: *U2UCF* produces exactly 10 recommendations for each user while KB_{Ut} recommends only 5 items on average. In contrast, in the notebook case the KB_{Ut} method produces 9 recommendations on average and therefore Precision is lower. When analyzing the notebook

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Table VII. Results on hybrids, $n = 10$

Dataset DC Strategy		$Prec$	Rec	$F1$	$Ccov$
$KB_{Ut}/U2UCF$	*cascade*	**6.8%**	8.97%	7.74%	42.17%
$U2UCF/KB_{Ut}$	*cascade*	**6.8%**	8.97%	7.74%	43.37%
$KB_{Ut}/U2UCF$	*weighted* (1:1)	3.73%	**23.14%**	6.42%	62.65%
$KB_{Ut}/U2UCF$	$switch_{KB_{Ut}rxw10}$	4%	**24.79%**	6.89%	62.65%
Dataset NB Strategy		$Prec$	Rec	$F1$	$Ccov$
$KB_{Ut}/U2UCF$	*cascade*	2.29%	3.75%	2.84%	26.84%
$U2UCF/KB_{Ut}$	*cascade*	3.05%	2.93%	2.99%	17.65%
$KB_{Ut}/U2UCF$	*weighted* 1.0/1.0	2.43%	**14.93%**	4.18%	70.22%
$KB_{Ut}/U2UCF$	$switch_{KB_{Ut}rxw10}$	2.52%	**15.52%**	4.34%	61.76%

case it turns out that not all explicit requirements can be satisfied for most users and therefore KB_{Ut} often has to relax filter constraints and thus produces more recommendations.

Unlike in the PC case, the cascading hybrids do not improve the results of the base algorithms (see TableVII) for DC and NB, neither in terms of Precision nor Recall. Although the DC cascade algorithms reach a remarkably high Precision compared to the other hybrid variants, it is still lower than the Precision reached by KB_{Ut} alone (see Table VI). However, the weighted algorithm variants that combine knowledge-based and collaborative recommendation lists outperform the base algorithms as was also evident in the PC dataset.

Due to the weak performance of the cascade versions, we applied our $switch_{m_1rxm_2N}$ switching strategy to the knowledge-based algorithm and the weighted hybrid. Once again, improvements in terms of Precision and Recall were observed with respect to the base and the hybrid techniques.

6. Discussion

The goal of these evaluation exercises was to compare the performance of different recommendation strategies in situations where only explicit customer requirements are available as the sole type of user feedback. Furthermore, different hybridization designs were explored to find more efficient algorithm configurations.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

As a consequence the results of these experiments must be seen in the specific context of providing personalized recommendations to anonymous or first-time online shoppers. As no item ratings are available for learning, standard experimentation designs would not be able to compute personalized results. Under these preconditions one successful recommendation was produced for each user of the cigar recommender out of a list with an average of 10 items by combining knowledge-based and collaborative methods. This strategy also paid off in the domains of digital cameras and notebooks by improving the results of base algorithms both in terms of Precision and Recall.

Consequently, short preference elicitation dialogues appear to be a valuable source of user feedback for personalization at no additional cost to the user. Parametric search forms that allow retrieval from a product catalog are part of nearly every commercial online platform. These experimental results indicate that it clearly makes sense to exploit user inputs to such search tools. Combining different types of feedback, for instance explicit user requirements and clickstream data like pageviews, remains part of the authors' ongoing research.

Knowledge-based recommendation paradigms have a clear advantage in utilizing explicit user input and transforming it into recommendations due to the encoded data semantics in their rules. Nevertheless, user-based collaborative filtering does astoundingly well in learning the relationships between requirements and items and comes to its full potential when hybridized with a method that exploits a thoroughly engineered knowledge-base. In contrast, an item-based collaborative filtering approach on explicit customer requirements achieved only dissatisfying results.

The utility-based technique clearly outperforms unpersonalized ranking schemes such as price. However, it is hard to encode more complex functional relationships within such schemes. Therefore, they are best used within a cascade hybrid for personalized rankings. Due to the high level of variance in the data, only a small number of functional dependencies between customers' requirements and product features could be mined with a high degree of confidence. However, the use of association rules as a validation mechanism for expert heuristics and rules is an additional interesting aspect that arose in the course of this study.

Nevertheless, the main contribution of this research lies in exploring different hybridization variants of knowledge-based and collaborative techniques for short-length user sessions on commercial websites. All three case studies showed that weighting and switching variants improved the initial recommendation strategies considerably. One reason for this behavior lies in the fact that knowledge-based and collaborative

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

methods produce successful recommendations (i.e. hits) for partially distinct sets of users. In our experiments $U2UCF$ did not only produce the identical hits as KB_{Ut} with some additional ones, but instead up to 50% of all successful recommendations derived by one method where not re-produced by the other method. The proposed switching hybrid is based on the heuristic to invoke the second method as a fallback strategy if the first one does not produce enough recommendations. Typically knowledge-based methods do not compute the maximal number of recommendations for each user due to their constraint-based reasoning mechanism. However, by switching to a weighted hybrid, as done in these experiments, ensures that *knowledge-based* product proposals are also part of the recommendation set of the fallback strategy. Still, even more adaptive switching criteria are on the agenda for future research. These could also take contextual parameters such as users' intentions or expectations into consideration for algorithm selection.

7. Conclusions

This paper placed emphasis on explicit customer requirements as a source of user feedback for personalization. The work was motivated by the special need to personalize the interaction with anonymous and first-time visitors. No data on buying transactions or item ratings is available for these customers and therefore they cannot be classified into user segments with similar tastes and preferences. This paper's contribution lies in instrumenting an evaluation on commercial data and in comparing all applicable recommendation strategies in that context. Furthermore, various hybridization designs for algorithms were explored and subsequently a new switching mechanism between knowledge-based and collaborative strategies was introduced that promises to overcome the identified shortcomings of each single technique. Summarizing, these experiments indicate, that user inputs to structured search forms or conversational recommender systems can be a valuable source of user feedback for personalization.

Acknowledgements

The authors would like to thank Arthur Pitman, Birgit Winkler and Marion Kollmann for proofreading and anonymous reviewers for their valuable and helpful comments.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

References

Adomavicius, G., R. Sankaranarayanan, S. Sen, and A. Tuzhilin: 2005, 'Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach'. *ACM Transactions on Information Systems* **23(1)**, 103–145.

Adomavicius, G. and A. Tuzhilin: 2001, 'Using Data Mining Methods to Build Customer Profiles'. *Computer* **34**(2), 74–82.

Adomavicius, G. and A. Tuzhilin: 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)**, 734–749.

Agrawal, R., T. Imielinski, and A. Swami: 1993, 'Mining association rules between sets of items in large databases'. In: *ACM SIGMOD Conference on Management of data*. Washington, D.C., USA, pp. 207–216.

Ardissono, L. and A. Goy: 2000, 'Tailoring the Interaction with Users in Web Stores'. *User Modeling and User-Adapted Interaction* **10**(4), 251–303.

Balabanovic, M.: 1997, 'An Adaptive Web Page Recommendation Service'. In: *1^{st} International Conference on Autonomous Agents*. Marina del Rey, CA, USA, pp. 378–385.

Balabanovic, M. and Y. Shoham: 1997, 'Fab: Content-Based, Collaborative Recommendation'. *Communications of the ACM* **40(3)**, 66–72.

Basilico, J. and T. Hofmann: 2004, 'Unifying collaborative and content-based filtering'. In: *21^{st} International Conference on Machine Learning (ICML)*. Banff, Alberta, Canada, pp. 9–16, ACM Press.

Berkovsky, S., T. Kuflik, and F. Ricci: 2008, 'Mediation of user models for enhanced personalization in recommender systems'. *User Modeling and User-Adapted Interaction* **18(3)**, to appear.

Breese, J., D. Heckerman, and C. Kadie: 1998, 'Empirical Analysis of Predictive Algorithms for Collaborative Filtering'. In: *14^{th} Conference on Uncertainty in Artificial Intelligence*. Madison, Wisonsin, USA, pp. 43–52.

Burke, R.: 2000a, 'Knowledge-based Recommender Systems'. *Encyclopedia of Library and Information Systems* **69(2)**, 180–200.

Burke, R.: 2000b, 'The Wasabi Personal Shopper: A Case-Based Recommender System'. In: *11^{th} Conference on Innovative Applications of Artificial Intelligence (IAAI)*. Trento, IT, pp. 844–849, AAAI.

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

Chu-Carrol, J. and S. Carberry: 1994, 'A Plan-Based Model for Response Generation in Collaborative Task-Oriented Dialogues'. In: *12^{th} National Conference on Artificial Intelligence(AAAI)*. Seattle, WA, USA, pp. 799–805.

Demiriz, A.: 2004, 'Enhancing Product Recommender Systems on Sparse Binary Data'. *Data Mining and Knowledge Discovery* **9**(2), 147–170.

Elzer, S., J. Chu-Carroll, and S. Carberry: 1994, 'Recognizing and Utilizing User Preferences in Collaborative Consultation Dialogues'. In: *4^{th} International Conference on User Modeling (UM)*. Hyannis, MA, USA, pp. 19–24.

Felfernig, A., G. Friedrich, D. Jannach, and M. Zanker: 2006, 'An Integrated Environment for the Development of Knowledge-Based Recommender Applications'. *International Journal of Electronic Commerce* **11**(2), 11–34.

Felfernig, A. and A. Kiener: 2005, 'Knowledge-based Interactive Selling of Financial Services using FSAdvisor'. In: *17^{th} Innovative Applications of Artificial Intelligence Conference (IAAI)*. Pittsburgh, PA, USA, pp. 1475–1482, AAAI Press.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Fogg, B. J.: 1999, 'Persuasive Technologies'. *Communications of the ACM* **42(5)**, 27–29.

Frakes, W. B. and R. Baeza-Yates (eds.): 1992, *Information Retrieval, Data Structure and Algorithms*. Englewood Cliffs, NJ, USA: Prentice Hall.

Goldberg, K., T. Roeder, D. Gupta, and C. Perkins: 2001, 'Eigentaste: A Constant Time Collaborative Filtering Algorithm'. *Information Retrieval* **4**(2), 133–151.

Goy, A., L. Ardissono, and G. Petrone: 2007, 'Personalization in E-Commerce Applications'. In: P. Brusilovsky, A. Kobsa, and W. Nejdl (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization*. Heidelberg, Germany, pp. 485–520, Springer.

Gretzel, U. and D. R. Fesenmaier: 2006, 'Persuasion in Recommender Systems'. *International Journal of Electronic Commerce* **11**(2), 81–100.

Harvey, T., S. Carberry, and K. Decker: 2005, 'Tailored Responses for Decision Support'. In: *10^{th} International Conference on User Modeling (UM)*. Edinburgh, Scotland, pp. 164–168.

Herlocker, J., J. Konstan, A. Borchers, and J. Riedl: 1999, 'An Algorithmic Framework for Performing Collaborative Filtering'. In: *22^{nd} International ACM Conference on Research and Development in Information Retrieval (SIGIR)*. Berkeley, CA, USA, pp. 230–237.

Herlocker, J. L., J. A. Konstan, L. G. Terveen, and J. T. Riedl: 2004, 'Evaluating Collaborative Filtering Recommender Systems'. *ACM Transactions on Information Systems* **22(1)**, 5–53.

Jannach, D.: 2004, 'Advisor suite - a knowledge-based sales advisory system'. In: L. S. Lopez de Mantaras (ed.): *16^{th} European Conference on Artificial Intelligence - Prestigious Applications of AI (PAIS)*. pp. 720–724, IOS Press.

Jannach, D.: 2006a, 'Finding Preferred Query Relaxations in Content-based Recommenders'. In: *IEEE Intelligent Systems Conference (IS)*. Westminster, UK, pp. 355–360, IEEE Press.

Jannach, D.: 2006b, 'Techniques for Fast Query Relaxation in Content-based Recommender Systems'. In: *29^{th} German Conference on Artificial Intelligence*. Bremen, Germany, pp. 49–63, Springer.

Jannach, D. and G. Kreutler: 2004, 'A Knowledge-Based Framework for the Rapid Development of Conversational Recommenders'. In: *5^{th} International Conference on Web Information Systems Engineering (WISE)*, Vol. LNCS 3306. Brisbane, Australia, pp. 390–402, Springer.

Jannach, D. and G. Kreutler: 2005, 'Personalized User Preference Elicitation for e-Services'. In: *IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE)*. Hong Kong, pp. 604–611.

Konstan, J. A., B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl: 1997, 'GroupLens: Applying Collaborative Filtering to Usenet News'. *Communications of the ACM* **40(3)**, 77–87.

Linden, G., S. Hanks, and N. Lesh: 1997, 'Interactive Assessment of User Preference Models: The Automated Travel Assistant'. In: *6^{th} International Conference on User Modeling (UM)*. Chia Laguna, Sardinia, Italy, pp. 67–78.

Linden, G., B. Smith, and J. York: 2003, 'Amazon.com Recommendations - Item-to-Item Collaborative Filtering'. *IEEE Internet Computing* **7**(1), 76–80.

McGinty, L. and B. Smyth: 2006, 'Adaptive Selection: An Analysis of Critiquing and Preference-Based Feedback in Conversational Recommender Systems'. *International Journal of Electronic Commerce* **11**(2), 35–57.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Mirzadeh, N., F. Ricci, and M. Bansal: 2004, 'Supporting user query relaxation in a recommender system'. In: *5th International Conference on E-Commerce and Web Technologies (EC-Web)*. Zaragoza, Spain, pp. 31–40, Springer.

Mobasher, B.: 2007, 'Data Mining for Web Personalization'. In: P. Brusilovsky, A. Kobsa, and W. Nejdl (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization*. Heidelberg, Germany, pp. 90–135, Springer.

Mobasher, B., R. Cooley, and J. Srivastava: 2000, 'Automatic personalization based on Web usage mining'. *Communications of the ACM* **43**(8), 142–151.

Pazzani, M.: 1999, 'A Framework for Collaborative, Content-Based and Demographic Filtering'. *Artificial Intelligence Review* **13**(5/6), 393–408.

Pazzani, M. J. and D. Billsus: 2007, 'Content-Based Recommendation Systems'. In: P. Brusilovsky, A. Kobsa, and W. Nejdl (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization*. Heidelberg, Germany, pp. 325–341, Springer.

Pierrakos, D., G. Paliouras, C. Papatheodorou, and C. D. Spyropoulos: 2003, 'Web Usage Mining as a Tool for Personalization: A Survey'. *User Modeling and User-Adapted Interaction* **13**(4), 311–372.

Pu, P. and B. Faltings: 2004, 'Decision Tradeoff Using Example-Critiquing and Constraint Programming'. *Constraints* **9**, 289–310.

Rafter, R. and B. Smyth: 2005, 'Conversational Collaborative Recommendation — An Experimental Analysis'. *Artifical Intelligence Review* **24**(3-4), 301–318.

Reilly, J., K. McCarthy, L. McGinty, and B. Smyth: 2005, 'Incremental critiquing'. *Knowledge-Based Systems* **18**, 143–151.

Reilly, J., J. Zhang, L. McGinty, P. Pu, and B. Smyth: 2007, 'Evaluating compound critiquing recommenders: a real-user study'. In: *8th ACM Conference on Electronic Commerce*. San Diego, California, USA, pp. 114–123, ACM Press.

Resnick, P., N. Iacovou, N. Suchak, P. Bergstrom, and J. Riedl: 1994, 'GroupLens: An Open Architecture for Collaborative Filtering of Netnews'. In: *Computer Supported Collaborative Work (CSCW)*. Chapel Hill, NC, USA, pp. 175–186.

Ricci, F.: 2002, 'Travel Recommender Systems'. *IEEE Intelligent Systems* **17(6)**, 55–57.

Ricci, F., A. Venturini, D. Cavada, N. Mirzadeh, D. Blaas, and M. Nones: 2003, 'Product recommendation with interactive query management and twofold similarity'. In: *5th International Conference on Case-Based Reasoning*. Trondheim, Norway, pp. 479–493, Springer.

Ricci, F. and H. Werthner: 2002, 'Case base querying for travel planning recommendation'. *Information Technology and Tourism* **3**, 215–266.

Riedl, J., J. Konstan, and E. Vrooman: 2002, *Word of Mouse: The Marketing Power of Collaborative Filtering*. New York: Warner Books.

Sacco, G. M.: 2000, 'Dynamic Taxonomies: A Model for Large Information Bases'. *IEEE Transactions on Knowledge and Data Engineering* **12**(3), 468–479.

Sarwar, B., G. Karypis, J. Konstan, and J. Riedl: 2000, 'Analysis of Recommendation Algorithms for E-Commerce'. In: *ACM Conference on e-Commerce (EC)*. Minneapolis, MN, USA, pp. 158–167.

Sarwar, B., G. Karypis, J. Konstan, and J. Riedl: 2001, 'Item-based Collaborative Filtering Recommendation Algorithms'. In: *10th International World Wide Web Conference*. Hong Kong, pp. 285–295.

Schafer, J. B., D. Frankowski, and J. H. S. Sen: 2007, 'Collaborative Filtering Recommender Systems'. In: P. Brusilovsky, A. Kobsa, and W. Nejdl (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization*. Heidelberg, Germany, pp. 291–324, Springer.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.): Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.

Schein, A. I., A. Popescul, D. M. Pennock, and L. H. Ungar: 2002, 'Methods and Metrics for Cold-Start Recommendations'. In: *25th International Conference on Research and Development in Information Retrieval (SIGIR)*. Tampere, Finland, pp. 253–260, ACM.

Shardanand, U. and P. Maes: 1995, 'Social information filtering: Algorithms for automating *word of mouth*'. In: *International Conference on Human Factors in Computing Systems (CHI)*. Denver, CO, USA, pp. 210–217.

Shimazu, H.: 2001, 'Expert Clerk: Navigating Shoppers' Buying Process with the Combination of Asking and Proposing'. In: *17th International Joint Conference on Artificial Intelligence (IJCAI)*. Seattle, WA, USA, pp. 1443–1448.

Smyth, B.: 2007, 'Case-Based Recommendation'. In: P. Brusilovsky, A. Kobsa, and W. Nejdl (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization*. Heidelberg, Germany, pp. 342–376, Springer.

Smyth, B. and P. Cotter: 2001, 'Personalized electronic program guides for digital TV'. *AI Magazine* **22**(2), 89–98.

Torrens, M., B. Faltings, and P. Pu: 2002, 'SmartClients: Constraint satisfaction as a paradigm for scaleable intelligent information systems'. *Constraints* **7**, 49–69.

Viappiani, P., B. Faltings, and P. Pu: 2006, 'Preference-based Search using Example-Critiquing with Suggestions'. *Artificial Intelligence Research* **27**, 465–503.

von Winterfeldt, D. and W. Edwards: 1986, *Decision Analysis and Behavioral Research*. Cambridge, UK: Cambridge University Press.

Wasfi, A. M. A.: 1999, 'Collecting user access patterns for building user profiles and collaborative filtering'. In: *4th International Conference on Intelligent User Interfaces (IUI)*. Los Angeles, CA, USA, pp. 57–64, ACM Press.

Witten, I. H. and E. Frank: 2005, *Data Mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann, 2nd edition.

Zanker, M., M. Bricman, S. Gordea, D. Jannach, and M. Jessenitschnig: 2006, 'Persuasive online-selling in quality & taste domains'. In: *7th International Conference on Electronic Commerce and Web Technologies (EC-Web)*. Krakow, Poland, pp. 51–60, Springer.

Zanker, M., M. Jessenitschnig, D. Jannach, and S. Gordea: 2007, 'Comparing Recommendation Strategies in a Commercial Context'. *IEEE Intelligent Systems* **22**(May/Jun), 69–73.

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems,
User Modeling and User-Adapted Interaction: The Journal of Personalization Research, A. Tuzhilin and B. Mobasher (Eds.):
Special issue on Data Mining for Personalization, Springer, Vol 19(1-2), 2009, pp.133-166.