Modeling User Preferences in Recommender Systems: A Classification Framework for Explicit and Implicit User Feedback

GAWESH JAWAHEER, City University London
PETER WELLER, City University London
PATTY KOSTKOVA, University College London (UCL)

Recommender systems are firmly established as a standard technology for assisting users with their choices; however, little attention has been paid to the application of the user model in recommender systems, particularly the variability and noise that are an intrinsic part of human behavior and activity. To enable recommender systems to suggest items that are useful to a particular user, it can be essential to understand the user and his or her interactions with the system. These interactions typically manifest themselves as explicit and implicit user feedback that provides the key indicators for modeling users' preferences for items and essential information for personalizing recommendations. In this article, we propose a classification framework for the use of explicit and implicit user feedback in recommender systems based on a set of dis- tinct properties that include Cognitive Effort, User Model, Scale of Measurement, and Domain Relevance. We develop a set of comparison criteria for explicit and implicit user feedback to emphasize the key properties. Using our framework, we provide a classification of recommender systems that have addressed questions about user feedback, and we review state-of-the-art techniques to improve such user feedback and thereby improve the performance of the recommender system. Finally, we formulate challenges for future research on improvement of user feedback.

Categories and Subject Descriptors: H.3.3 [**Information Systems–Information Storage and Retrieval–Information Search and Retrieval**]: Information Filtering; A.1 [**General Literature**]: Introductory and Survey

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Feedback, recommender systems, explicit feedback, implicit feedback, improvement of feedback

ACM Reference Format:
Gawesh Jawaheer, Peter Weller, and Patty Kostkova. 2014. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. ACM Trans. Interact. Intell. Syst. 4, 2, Article 8 (May 2014), 26 pages.
DOI: http://dx.doi.org/10.1145/2512208

1. INTRODUCTION

With the overwhelming information on the Internet and limitations of one-fit-all search engines, advanced tools are required to enable users to find the right information and make choices meeting their needs and expectations, thus enhancing their engagement and overall satisfaction with online services. Recently, recommender systems have been increasingly popular in assisting users with their choices. A recommender system can be abstracted to consist of a user model, a community, an item (product) model, a

Author's addresses: G. Jawaheer, Centre for Health Informatics, School of Informatics, City University Northampton Square, London EC1V 0HB, UK; email: Gawesh.Jawaheer.2@city.ac.uk.

recommender algorithm, and an interaction style [Zanker and Jessenitschnig 2009]. The user model provides all of the information for personalizing the user's experience. It captures the user interactions with items in user profiles. Mainly, these user interactions consist of explicit and implicit information about the user's interest or preference for items. Typically, recommender systems use ratings as a mechanism to proactively express their interests in items and seamlessly collected clickstream data for inferring users' interests or preferences. This explicit and implicit information are usually referred to as explicit feedback or explicit rating and implicit feedback, or implicit rating [Konstan et al. 1997; Jannach et al. 2011]. There has been significant research activity in this area since the 1990s. However, relatively little attention has been given to questioning how user feedback is applied to recommender systems. Several recommendation algorithms do not account for the variability in human behavior and activity. Often, they are hardwired for explicit ratings rather than implicit ratings.

User feedback is an indispensable part of most recommender systems. Thus, studying user feedback can have a profound impact on recommender systems' technology and understanding of the user. As an illustration, Amatriain et al. [2009b] showed that a simple strategy of removing the noise in 20% of explicit ratings can improve the root mean square error (RMSE) of the predictions made by the recommender system by more than 5%. In this research field, much emphasis has been laid on algorithmic improvements of recommender systems. However, recently, there has been a growing body of work looking at other aspects of recommender systems; in particular, the issues around explicit and implicit user feedback have received prominent attention [Amatriain et al. 2009a; Parra and Amatriain 2011; Parra et al. 2011; Koren and Sill 2011; Hu et al. 2008].

Improvements in the user feedback of recommender systems can be an efficient way to enhance their performances across a wider domain of input data compared to purely algorithmic improvements due to the pervasiveness of user feedback in recommender systems and the fact that algorithmic improvements largely depend on the area under study, particularly the dataset used [Huang 2007; Herlocker et al. 2004]. For example, researchers have shown that a technique like re-rating can be a more efficient way of improving the overall performance of recommender systems compared to purely algorithmic variants [Amatriain et al. 2009b].

In this article, we propose a classification framework for the use of explicit and implicit user feedback in recommender systems based on a set of distinct properties. According to this framework, we classify recommender systems by utilizing explicit and implicit user feedback as key indicators for modeling users' preferences and com- pare techniques of improving the elicitation and application of user feedback to recommender systems. We focus on recommender systems' use of user feedback for the purpose of this article rather than considering all literature on user feedback. How- ever, we include overlapping research especially in the area of implicit user feedback in information retrieval. In addition, most of research reviewed falls into collaborative filtering-based recommender systems. We describe ways to elicit user feedback and discuss its application in recommender systems to enhance their performance. According to our classification framework, we discuss the distinct properties of each form of user feedback and provide a comparison in terms of their properties, key differences, and similarities. We also suggest the ways to improve user feedback in recommender systems and present a comparison of state-of-the-art techniques. Finally, we describe some challenges in the area and future directions.

2. DEFINITIONS, NOTATIONS, AND DATASETS

We start with some definitions and notations relevant to the domain. We will also briefly describe the datasets that are commonly used by researchers and discuss several issues related to datasets, particularly the scarcity of implicit feedback datasets.

2.1. Definitions and Notations

We will use the generic formal definition of recommender systems given by Adomavicius and Tuzhilin [2005]. Given that we have a user $u \in U$, where U is the set of all users, and an item $i \in I$, where I is the set of all items that can be recommended. Assume that we have a utility function that measures the usefulness or interest of an item i to user u such that

$$\mu : U \times I \rightarrow R. \tag{1}$$

Information about the users and items can be stored in user and item profiles, respectively. In recommender systems, the utility function μ is typically known over only a subset of space $U \times I$, and thus the problem in recommender system is to extrapolate the utility function μ over the space $U \times I$ [Adomavicius and Tuzhilin 2005]. Typically, the utility function maps the user-item matrix to explicit user ratings. Let explicit user feedback and implicit user feedback between user u and item i be expressed as $f^{exp}_{u,i}$ and $f^{imp}_{u,i}$, respectively. Hence, for a more generic definition that includes implicit feedback, we assume that

$$R = \mu\left(f^{exp}_{u,i}, f^{imp}_{u,i}\right). \tag{2}$$

Recently, we have seen research that uses both forms of feedback in recommender systems [Parra and Amatriain 2011; Parra et al. 2011; Koren 2010]. However, traditionally, most recommender systems have either used one or the other form of user feedback. This could be due to the lack of publicly available datasets that contain both explicit and implicit user feedback.

Note that in this article we employ a simple user model. A useful augmentation is considering the context of the user feedback [Ricci et al. 2010; Adomavicius et al. 2011]—that is,

$$\mu_c : U \times I \times C \rightarrow R, \tag{3}$$

where C represents the contextual factors [Adomavicius et al. 2011] that affects the user feedback and hence the recommendations. The notion of context in recommender systems is outside the scope of this article, and instead we point readers to Ricci et al. [2010] for references.

2.2. Datasets

Recommender systems are very useful when applied to assisting users with choices for online items, typically music, books, movies, or resources. In fact, as long as there is data captured about the interactions of users with an item and some notion of preference between users and items, the use of a recommender system becomes a theoretical possibility. For research purposes, however, publicly available datasets are actively used for empirical offline evaluations of recommender systems. But datasets should be used with caution. Herlocker et al. [2004] warned of the inappropriate use of datasets, stressing that evaluations will be meaningful if the dataset has the same characteristics of the target recommender system, such as the ratio of users over the items, the rating density, the rating sparsity, and the rating scale. Generalization of the results of offline evaluations of recommendation algorithms must be scrupulously analyzed.

A trio of datasets consisting of explicit user ratings of movies appears very often in recommender systems literature, namely the Netflix, MovieLens, and Each- Movie datasets. Although still used by researchers, the Netflix [Bennett and Lanning 2007] and the EachMovie datasets [GroupLens Research 2011a] are no longer publicly available. The MovieLens dataset is divided into three parts: 100,000 movie ratings (943 users and 1,682 movies), 1 million movie ratings (6,040 users and 3,900 movies), and 10 million movie ratings (71,567 users and 10,681 movies) [GroupLens Research 2011b]. Ratings in the MovieLens dataset are on a 5-point integer scale. While the use of these datasets enables comparative studies, it limits the generalizations of results to other domains.

As for implicit user feedback, a popular and publicly available dataset is the Last.fm dataset [Herrada 2009a], which contains the full playing history for about 1,000 users and the total number of artist playcounts (the number of times an artist's album was played) for about 360,000 users from the Last.fm online music recommender system.

There is only one publicly available dataset having both explicit user feedback and implicit user feedback, namely the book-crossing dataset [Ziegler 2004], which includes 1,149,780 explicit and implicit ratings from 278,858 users on 271,379 books. Explicit ratings in the book-crossing dataset are on a 1 to 10 integer scale, whereas implicit user feedback is expressed with the value of zero.

Another potential source of explicit and implicit user feedback in a recommender system is Last.fm. Although the publicly available dataset mentioned previously [Herrada 2009a] includes only implicit user feedback, using the Last.fm API, one gets access to both explicit user feedback (binary ratings—users can "love" or "ban" tracks) and im- plicit user feedback (playcount). Our previous works have collected both implicit and explicit user feedback using the Last.fm API [Jawaheer et al. 2010a, 2010b]. It is our hope to make our dataset publicly available in the future.

Another approach for the lack of datasets that include both explicit and implicit user feedback is to convert explicit ratings into pseudoimplicit ratings. This approach was taken by Koren [2010]. He used the Netflix dataset, which contains movie ratings by users as explicit user feedback [Bennett and Lanning 2007]. In regard to implicit user feedback, he used a binary value to represent whether a user has rated a movie or not (i.e., pseudoimplicit ratings). The reasoning behind this approach is that a user rating a movie is not a random act, and thus the user is implicitly providing some information about her or his preferences [Marlin and Zemel 2009]. Hence, any dataset with explicit user feedback also contains a pseudoimplicit user feedback counterpart with binary values. But such implicit user feedback is not very rich. Nevertheless, Koren [2010] found that incorporating this naive implicit user feedback with explicit user feedback increases the prediction accuracy compared to solely using explicit user feedback.

Using datasets provides quick and repeatable evaluations of recommender systems. Such evaluations using explicit user feedback are well researched with some defined metrics [Herlocker et al. 2004]. On the other hand, evaluations of recommender systems using implicit user feedback are still under research and development. We will revisit the issue of evaluation in Section 7.5.

3. EXPLICIT USER FEEDBACK IN RECOMMENDER SYSTEMS

Although implicit user feedback can be seamlessly collected and seems a natural candidate for modeling user online behavior and preferences, recommender systems research has predominantly focused on explicit feedback. Explicitly asking the user to rate an item using a scale has become the de facto way of expressing user interest about an item in recommender systems. In the following sections, we will give examples of the use of ratings in recommender systems and discuss reliability and other issues affect- ing its use. Finally, we will discuss other ways of capturing explicit user feedback in recommender systems.

3.1. Ratings

Typically, to elicit explicit user feedback, recommender systems have used an N point Likert response scale. Points on the scale are converted to numerical values representing user preferences [Jannach et al. 2011]. There are different rating interfaces that can be used. The MovieLens recommender system allowed users to rate movies on a 5-point scale [GroupLens Research 2011b]. The Jester Joke recommender system used a continuous scale of real values from −10 to 10 +[Goldberg et al. 2001; Jester Dataset 2012]. Ratings can also have binary values. Last.fm allows users to express explicit user feedback through a binary rating; a user can either "Love a track" (positive user feedback) or "Ban a track" (negative user feedback) [Jawaheer et al. 2010a]. Binary ratings are also available in other systems like Facebook or YouTube [Davidson et al. 2010]. Interfaces can also capture unary ratings (i.e., only positive ratings) such as the "Favorite" feature on Twitter.

Several researchers have investigated the use of ratings in recommender systems. Cosley et al. [2003] studied the reliability of ratings - we will discuss this in Section 3.1.1 -, the influence of showing rating predictions at the time of asking users to rate and the granularity of the rating interface used. Cosley et al. [2003] showed statistically significant results that users re-rated more often at their original ratings when they were shown the predictions. It could be that showing the predictions helped users remember their ratings or they were influenced by the predictions. They showed that the latter hypothesis was more plausible as statistical significant results showed that when users were presented with deliberately incorrect predictions (1 star), ±they rated above or below their original rating more often than when they are presented with original ratings [Cosley et al. 2003].

Cosley et al. [2003] confirmed their hypothesis that users are influenced by showing predictions with another experiment where users were shown movies that they have not rated before. An experimental group was presented with ratings without predic- tions, ratings with predictions, and ratings with manipulated predictions at the 1 ± level. In a control group, users were shown ratings with the actual predictions. The two groups were also subject to satisfaction surveys. Statistically significant results showed that altering the predictions downward caused users to rate lower than when actual predictions were shown and that altering the predictions upward caused users to rate higher than when actual predictions were shown. In addition, users rated their predictions more often compared to those when they were not shown any predictions. These results provided further evidence that showing predictions influenced the users. However, the satisfaction survey showed that users in the control group were more satisfied than the experimental group, leading the researchers to conclude that users could detect that the predictions were not accurate.

In another experiment, Cosley et al. [2003] asked users to re-rate movies in different scales than their original ratings (1 to 5 levels). Three scales were used for re-rating, namely binary, −3 to +3 without zero, and half-level (0.5 to 5 levels in increments of 0.5). Users were also subjected to a satisfaction survey. From the satisfaction survey, the order of preference of the rating scales was as follows: the most liked scale being the half-level scale, then the non-zero −3 to+3 scale, and finally the binary scale. However, all ratings on the three scales correlate strongly with the original ratings, meaning that they all are useful. However, the researchers were unable to provide conclusive evidence between the choice of scales and the performance of the predictions made by the recommender system.

But which best rating interface to use for recommender systems is an open question. Goldberg et al. [2001] employed a continuous scale for the Jester Joke recommender system because it provides finer granularity and argued that it avoids the loss of information when a discrete scale is converted to a scalar value. Cosley et al. [2003] also found that users prefer finer-grain scales. However, whether finer scales lead to an improvement in the recommendations is still an open question, as Cosley et al. [2003] noted that their findings on this aspect were not conclusive. Recent research has shown that rating with finer granularity puts higher cognitive load on the user and that better user satisfaction is observed with binary or five-star ratings rather than unary or slider scales ratings [Sparling and Sen 2011]. Kluver et al. [2012] proposed an information theoretic model for studying the preference information in ratings and their predictions. The model provides a means for comparing different rating interfaces. However, considering that they evaluated their model on a synthetic dataset, they acknowledge that further work is required to evaluate its performance on natural datasets.

3.1.1. Reliability of Ratings. Cosley et al. [2003] also studied the reliability of ratings. Their experiments on re-rating described in the previous section showed that users re-rated to their original rating 60% of the time and that the correlation between the original ratings and the re-rated values was 0.70. A similar study of re-rating by Hill et al. [1995] used an email interface to recommend movies and acquire ratings. Users were asked to re-rate the same list of movies after a 6-week gap. The Pearson correlation coefficient between the original and re-rated list was 0.83. This higher correlation, compared to that of Cosley et al. [2003], could be because Cosley et al. [2003] restricted re-ratings only to movies with original ratings in the middle of the range (2 to 4 integer values); middle ratings have twice as much chance to re-rate at different values than extreme ratings (1 or 5). One shortcoming of the work by Cosley et al. [2003] is that they did not consider the time gap between the original ratings and the re-ratings. Thus, they did not isolate the effect of the user's memory on the re-ratings. Nevertheless, the results showed that ratings have a certain level of uncertainty.

O'Mahony et al. [2006] investigated noise in ratings and how to eliminate such noise. They distinguished between natural noise and malicious noise. Their work was based around the notion of the consistency of ratings defined as the mean absolute error between the actual and predicted rating. Ratings above a threshold were classed as noise and removed from the recommendation process. The threshold can be regarded as the value within which normalized predicted ratings are allowed to vary from the normalized actual ratings. We will review and discuss their findings in Section 6.1.

Amatriain et al. [2009a] studied the reliability of ratings. They performed experiments asking users to rate and re-rate movies randomly selected from the Netflix dataset [Bennett and Lanning 2007]. In contrast to the work by Cosley et al. [2003], the researchers tried to isolate the effects of the time gap between the original ratings and the re-ratings. The researchers used the test-retest method from classical test theory to estimate reliability. As they explained, in a test-retest method, two ratings may be different because of the reliability of the instrument or the stability of the user's judgment. Thus, the researchers computed estimation for reliability and stability. They found that using rating for movies was a reliable metric for judging the like and dislike of movies (overall reliability of 0.93). However, this does not mean that it is also a reliable metric for conferring user preference. They found that there was one anomaly on the stability measures; in one pair of experiments, the lowest stability did not correspond to the longest time interval between the experiments. This anomaly suggests that other factors could be involved in stability. They also found that extreme ratings in the scale have a greater influence of the reliability rather than middle ratings. They demonstrated *that user ratings inherently had noise*: users were unable to distinguish movies that they had seen or not even if the re-rating experiment was separated by 1 day. They found that the ratings in the middle of the rating scale are more prone to inconsistencies. The order of presentation of the movies was shown to have an effect on inconsistencies, as grouping movies with similar likelihood of receiving a rating will reduce the inconsistencies. They also showed that the speed of providing rating does not have an effect on inconsistency. Finally, they found that rating does not provide an appropriate way to measure user interest over the long term.

The most important aspect of the preceding reviewed research is that ratings should not be considered as absolute truth. But most recommendation algorithms consider ratings as such. We will further expand on this when we discuss the characteristics of user feedback in Section 5.

3.1.2. Applications of User Ratings in Recommendation Algorithms. Recommender systems literature shows that ratings are the de facto means of eliciting explicit user feedback. Several recommendation algorithms are hardwired to use ratings for computing recommendations. But as seen in the previous section, ratings should not be considered as the absolute truth. However, most recommendation algorithms do not account for this unreliability. In a broader sense, there have been questions raised about whether the research community has been using ratings in the correct way in recommendation algorithms. In an article by Robertson [2011], it was stated that although ratings are ordinal data, 92% of papers presented at the CHI 2009 Conference considered rating as interval data rather than ordinal data. This finding has implications in recommender systems, as several similarity measures and performance metrics like RMSE also re- gard rating data as being interval data rather than ordinal data [Amatriain 2011].

Koren and Sill [2011] have argued that although expressing user feedback (both explicit and implicit) as numerical values is intuitive, it is not a natural representation because it limits the expressiveness of the user feedback and is hard to quantify the various levels of user feedback. They also argued that each individual has different internal scales on which they rate items of interest, and by using ratings as absolute numerical preferences, we lose information about these individual scales. Hence, they proposed a framework called *OrdRec*, which models user feedback as ordinal data. Its implementation uses matrix factorization referred to as SVD++ [Ricci et al. 2010] but with the ratings as ordinal data. They evaluated their framework on standard datasets using RMSE and a metric that considers the ranking of the ratings. They found different relative performances between OrdRec and the baselines depending on the datasets they used. Recently, other researchers have also modeled ratings as ordinal data [Parra et al. 2011].

3.2. Other Means of Explicit User Feedback in Recommender Systems

Although ratings have been established as the key method for eliciting explicit user feedback, we now describe other possible means of explicit user feedback.

There is literature on the use of comments and product reviews as explicit user feedback that helps the community of users [Leino and Raiha 2007; Lu et al. 2009; Siersdorfer et al. 2010], and there is literature on how such explicit user feedback can be applied to recommender systems [Garcia Esparza et al. 2012; Aciar et al. 2006]. Furthermore, Desrosiers and Karypis [2010] proposed an alternative algorithm for computing similarities—a key function of collaborative-based recommender systems—using nonnumerical ratings. In the context of conversational recommender systems, two techniques have been proposed as alternatives to ratings, namely critiquing [McGinty and Smyth 2005] and preference-based user feedback [McGinty and Smyth 2002]. The notion of user feedback in conversational recommender systems is close to the informa- tion retrieval domain. For example, conversational recommender systems start with a query and then employ user feedback to refine this query and eventually the recommen- dations. Despite these attempts, ratings remain a prominent way of eliciting explicit user feedback for recommender systems and cannot be substituted by critiquing or preference-based user feedback in collaborative-based, content-based, or hybrid rec- ommender systems without major changes to the interface and the algorithms.

Szomszor et al. [2007] have used tagging as an additional source of explicit user feedback in content-based recommendations. Their approach was motivated by the assertion that the performance of recommender systems can be improved by combining data from different sources to build richer profiles. Their experiments consisted of improving recommendations using the Netflix dataset by harvesting tag clouds from IMDB. They used an ontological framework to combine and query the two datasets. They evaluated their algorithms over 500 randomly chosen users from the Netflix dataset and got promising results over the baselines. However, these results cannot be generalized, because they evaluated their algorithms on a very small sample (0.10% of the users) of the Netflix dataset (480,189 users). Nevertheless, it provides encouraging results for using other types of explicit user feedback in recommender systems.

3.3. Summary

The preceding sections demonstrate that recommendation algorithms should not consider ratings as the absolute truth about user choices and cannot be relied on 100% for modeling user preferences and decision-making process. Nevertheless, there is paucity in research that challenged the use of ratings in recommender systems. Converting the user’s explicit interest—in itself a complex concept—into a numerical value causes loss of information [Koren and Sill 2011]. Recommender systems research has been focused on purely new algorithms or improvement of existing algorithms. Thus, in this context, ratings provide researchers with an intuitive and simple process of using explicit user feedback in recommendations while they concentrate on other aspects of the algorithm. Furthermore, evaluating the performance of recommendation algorithms through prediction of ratings is easier and more reliable than evaluating the unknown preferences of users. But recent research by Koren and Sill [2011] showed that there is other less intuitive but more principled ways of using ratings in recommender systems.

Following this review of the literature on using explicit user feedback in recommender systems, we now discuss the key points in applying implicit user feedback in recommender systems.

4. IMPLICIT USER FEEDBACK IN RECOMMENDER SYSTEMS

Although widely available and seamlessly collected in recommender systems research, implicit feedback is considered secondary to explicit user feedback in such research. The recommender systems literature is focused on explicit user feedback rather than implicit user feedback [Hu et al. 2008]. However, Adomavicius and Tuzhilin [2005] argued that future recommender systems will need to be less intrusive, thus relying more on implicit user feedback to provide recommendations.

Implicit user feedback is based on observable behaviors exhibited by a user. Nichols [1997] first surveyed a list of useful behaviors. Later, Oard [1998] extended this list, building a framework to categorize these behaviors into three sets, namely *examination*, *retention* and *reference*. Finally, Oard and Kim [2001] added a further refinement by adding *annotation* as an additional category and breaking down these observable behaviors based on the scale at which the observations were made, as shown in Table I. Detailed explanations and justifications of these behaviors are available in Oard and Kim [2001].

The addition of the Annotate category can be seen as a way of unifying explicit user feedback and implicit user feedback onto the same framework, as the Rate and Publish behaviors are associated with explicit user feedback [Oard and Kim 2001]. This is useful because it provides a means of discussing all categories of user feedback in recommender systems by using a single theme of observable behaviors. Based on this approach, we introduced the “scale” variable in line with our feedback unification paradigm.

There are a number of studies within the recommender systems literature that looked at the observable behaviors relating to user preference. Konstan et al. [1997], who

Table I. Matrix of Observable Behaviors

Behavior Category	Scale		
	Segment	Object	Class
Examine	View Listen	Select	
Retain	Print	Bookmark Save Delete Purchase	Subscribe
Reference	Copy/paste Quote	Forward Reply Link Cite	
Annotate	Mark up	Rate Publish	Organize

implemented and evaluated a collaborative filtering system applied to Usenet news, found that the time spent reading (implicit user feedback) was correlated with rating (explicit user feedback), where the greater the time spent reading meant the higher the ratings. The authors claimed that predictions made using such implicit user feedback were as accurate as predictions made using explicit user feedback. Unfortunately, they did not provide any quantitative measure to justify their claim. The researchers also found that implicit user feedback could be part of a solution to the "early adopter problem," which meant that users need to rate articles to see the benefit of the system and that only a few articles would get ratings in the beginning, making predictions available for only a few articles. This line of research requires further investigation.

Claypool et al. [2001] have found that the time spent on a Web page ("examine" behavior) is a statistical indicator of interest and was linearly proportional to explicit rating of interest. Their research also found that scrolling was useful as an indicator of interest but that it did have a linear proportional relationship with interest, whereas mouse clicks were found not to be an indicator of interest. This research by Claypool et al. [2001] is often quoted in the recommender systems literature. However, it is important to analyze the research in context. The experiments from this research were carried out using a special browser that measured several user behaviors on the client side, whereas most of the recommender systems had been implemented on the server side. In terms of coverage, implicit interest indicators based on reading time had less coverage (70%) on the server side than on the client side (100%), which in turn had more coverage than the explicit interest indicators (only 80% of Web pages). Implicit interest indicators on the server side have less coverage than on the client side, because on the server side, reading time can only be computed within a session using the access time for consecutive pairs of Web pages, making it impossible to compute the reading time for the last page. The researchers assumed that explicit interest indicators are 100% accurate; based on this assumption, they calculated the client-side implicit interest indicators to be 70% accurate. Hence, combining the two metrics of accuracy and coverage, explicit interest indicators had 80% accurate coverage, client- side implicit interest indicators had 70% accurate coverage, and server-side implicit interest indicators had 50% accurate coverage.

As the researchers pointed out, this experiment was conducted in a controlled environment, whereas in normal conditions, the users would have more distractions, thus possibly making the correlation between time spent on a page and interest less strong. In addition, this research was done in 2001 using Internet Explorer. Since then, there have been new browser features such as tabbed browsing, which makes it even more unreliable to calculate the time spent on a page. Furthermore, another limitation is that the researchers assumed the explicit interest indicator to be 100% accurate, which in turn was used to calculate both the client-side and server-side implicit interest indicators. Thus, the researchers failed to take the user variance of ratings into consideration. For example, although the works done by Cosley et al. [2003], Hill et al. [1995], and Amatriain et al. [2009a] were in different domains (rating movies vs. Web pages), they nevertheless showed that explicit user feedback was not 100% accurate.

Kim and Oard [2001] conducted two experiments between reading time and explicit relevance judgments among students who are reading journal articles. They found that the mean reading time generally increases with explicit high relevance. However, it was not a proportional increase: in one experiment, the articles with a “moderate interest” rating had the highest mean reading time, whereas in the other experiment, the articles with “low interest” had the highest rating, followed by the moderate interest rating and high interest rating, respectively. Based on this result, the researchers argued that the reading time was not able to distinguish between degrees of interest while being able to distinguish between relevant and nonrelevant articles (relevant articles had higher mean reading time than nonrelevant articles), suggesting that implicit user feedback like reading time can only be binary. A potential bias in these experiments is that the number of relevant articles to nonrelevant articles was 5:1 rather than 1:1. Another point is that using median rather than the mean time reading may have been a better way of comparing reading time against interest. Claypool et al. [2001] used median reading time because they observed that the outliers make the median a better statistic than the mean.

4.1. Applications of Implicit Feedback in Recommendation Algorithms

Although researchers in recommender systems have long been interested in implicit user feedback, there is relatively little published research on implementation and design of algorithms that use implicit user feedback for generating recommendations. This is likely due to the lack of datasets with implicit user feedback. On the other hand, datasets like MovieLens or Netflix have galvanized research on recommender systems that process explicit user feedback. However, recently we have seen several papers that deal with the implementation of recommendation algorithms using implicit user feedback (e.g., namely Gadanho and Lhuillier [2007], Hu et al. [2008], Koren [2010], Liu et al. [2010], Parra and Amatriain [2011], Parra et al. [2011], Moling et al. [2012], and Rendle et al. [2009]). To be consistent with the structure of this article, we will only review the papers by Hu et al. [2008] and Rendle et al. [2009] in this section, as they solely process implicit user feedback. We will review the rest of the papers [Gadanho and Lhuillier 2007; Koren 2010; Liu et al. 2010; Parra and Amatriain 2011; Parra et al. 2011; Moling et al. 2012] in Section 6, discussing improvement of user feedback— specifically, the techniques of improving explicit user feedback by using implicit user feedback in Section 6.5 and improving the uncertainty in implicit user feedback in Section 6.4.

Hu et. al. [2008] argued that to use implicit user feedback as user preference in a recommender system, we need to convert the expressed confidence into user preference. The implication is that explicit user feedback quantifies a user’s interest in an item, whereas implicit user feedback can only approximate the confidence of the user’s interest—that is, explicit and implicit user feedback are not directly comparable.

Hu et al. [2008] suggested mapping observable behaviors r_{ui} as an expression of confidence c_{ui} using

$$c_{ui} \left(1 \quad \alpha log \right) 1 \quad \frac{r_{ui}}{+\varepsilon} \quad . \tag{4}$$

This expression is not prescriptive and would most likely depend on the domain and the observed behaviors. Hu et al. [2008] also suggested that a simple strategy of setting a binary user preference could be to set user preference $p_{ui} = 1$ when observable behaviors $r_{ui} > 0$ and p_{ui} 0 otherwise. Other researchers have also suggested using binary user preference values when it comes to implicit feedback [Kim and Oard 2001; Koren 2010]. For example Kim and Oard [2001] claimed that using reading time as an observable behavior in an implicit user feedback recommender system was not able to go beyond binary user preference. The approach taken by Hu et al. [2008] is to weigh the binary preference by c_{ui}. Hu et al. [2008] evaluated their algorithms using a private dataset that they used for building a TV show recommender system. In their case, expressions of implicit user feedback, r_{ui}, referred to how many times a user watched a particular show. They implemented a latent factor model based on the implicit user preference p_{ui} weighted by the confidence c_{ui}. As baselines, they implemented an item-based neighborhood model and a model based on recommending shows solely based on popularity. Their results showed that latent factor model based on confidence performed better than the baselines. Unfortunately, as their evaluations of performance of the three models used a domain-dependent metric of the ranking of the shows, we cannot generalize these findings. In fact, having a standard metric for comparing the accuracy of predictions across different experiments is still a challenge in the context of recommender systems that process implicit user feedback.

Rendle et al. [2009] model item recommendation using implicit user feedback as the prediction of a personalized ranking on a set of items. In fact, modeling item recommendation as personalized ranking has the benefit of opening a gamut of machine learning techniques than can be applied to the problem. Rendle et al. [2009] provided a Bayesian solution to the ranking problem that they called *Bayesian personalized ranking* (BPR). They apply BPR to two common recommendation algorithms, namely adaptive kNN and matrix factorization. They evaluate their model using a private dataset of Web- based transactions of an online shop and a sample of the Netflix dataset. But as the latter does not contain implicit user feedback, the researchers converted explicit rat- ings into pseudoimplicit ratings as described in Section 2.2. They used the area under the curve (AUC) as an evaluation metric [Herlocker et al. 2004]. SVD and weighted regularized versions of matrix factorization and cosine-kNN served as baselines. Their evaluation results show that the BPR versions of both matrix factorization and kNN outperformed the other versions.

4.2. Summary

In this section, we described how implicit user feedback can be used in recommender systems. However, despite interest in this area, we note that there is paucity in research that implements recommendation algorithms that process implicit user feedback. In the following section, we now discuss the various characteristics of the two forms of user feedback to highlight their key pertinent differences.

5. CHARACTERISTICS OF USER FEEDBACK IN RECOMMENDER SYSTEMS

It is crucial to identify the characteristics of the two forms of user feedback in recommender systems in order to make the right design choices when having to choose between explicit user feedback and implicit user feedback or both. In addition, Hu et al. [2008] highlighted the fact that it is important to identify the unique characteristics of implicit user feedback that prevent the direct use of recommendation algorithms that have been designed for explicit user feedback.

To help us discuss these two forms of user feedback with a view of highlighting the key aspects, in our framework we developed a set of comparison criteria for implicit and explicit feedback as listed in Table II.

Cognitive Effort: Acquiring implicit user feedback is seamless, whereas explicit user feedback requires some cognitive effort [Gadanho and Lhuillier 2007]. This is one of the

Table II. Properties of User Feedback in Recommender Systems

Properties	Explicit User Feedback	Implicit User Feedback
Cognitive effort	Yes	No
User model	Preference	Confidence
Scale of measurement	Ordinal	Ratio
Domain relevance	Irrelevant	Relevant
Sensitivity to noise	Yes	Yes
Polarity	Positive and negative	Positive
Range of users	Subset of users	All users
User transparency	Yes	No
Bias	Power users	No bias

reasons only a small percentage of users contribute explicit user feedback. However, the motivations to explain why users provide explicit user feedback are complex. In a study of MovieLens users, Harper et al. [2005] found that users rate movies for a variety of reasons, namely to make a list for themselves, to influence others, for their enjoyment of the activity, or because they think that it improves their recommendations. Improving our understanding of the explicit behavior of users in recommender systems can help us improve recommendations.

User Model: As we mentioned in Section 2, we employed a simple user model without contextual factors. In this case, we highlight only one aspect of the user model, namely how we represent user feedback in the recommender system. Hu et al. [2008] argued that the numerical values of explicit user feedback denote user preference, whereas numerical values of implicit user feedback denote confidence. In the case of explicit user feedback, converting ratings as numerical values of user preference is intuitive, albeit with some information loss as Koren and Sill [2011] argued. In the case of implicit user feedback, the recommender system must interpret observable behaviors. The school of thought adopted by Hu et al. [2008] is that there are many reasons a user may behave in a particular way. However, recent work by Parra and Amatriain [2011] and Parra et al. [2011] modeled implicit feedback as explicit user ratings (i.e., user preferences) using linear regression and ordinal regression. This provides a counterargument to the view of implicit feedback by Hu et al. [2008]. Another school of thought is that preference is complex and converting user preference to a numerical value is nontrivial [Lichtenstein and Slovic 2006]. There is substantial literature in psychology on the notion of preference. According to one school of thought, users do not really know what they prefer; instead, they construct their preferences as the situation evolves [Lichtenstein and Slovic 2006].

Scale of Measurement: Explicit feedback is usually expressed as ratings (e.g., Likert scale) that have an ordinal scale [Field and Hole 2003]. Implicit feedback is typically measured as some form of counting of repeatable behaviors. For example, in a music recommender system, implicit user feedback could be the number of times that a track was played. Thus, typically implicit user feedback will have a ratio scale of measurement. This makes it difficult to include explicit and implicit feedback in the same user model [Liu et al. 2010]. Furthermore, having different scales of measurement makes it impossible to compare explicit and implicit feedback. For example, given the option, should we choose explicit feedback or implicit feedback when building a recommender system? Recent work by Kluver et al. [2012] has addressed this problem. Their information theoretic model can be used to compare different scales of measurement, as it uses the same scale to measure explicit and implicit feedback, namely the preference information in ratings and their predictions.

Domain Relevance: Interpretation of explicit user feedback is irrelevant to the domain under study, whereas domain knowledge is essential to interpret implicit user feedback.

Sensitivity to Noise: Intuitively, by its nature, implicit user feedback is sensitive to noise [Gadanho and Lhuillier 2007]. Noise in implicit user feedback may be caused by the user, the model used for inferring the user preference, and the system noise, which includes noise generated by the tools used for capturing the user feedback. Amatriain et al. [2009a] have shown that explicit user feedback is also sensitive to noise. In the latter case, noise in explicit user feedback is caused by the user and the system.

Polarity: Explicit user feedback can be positive and negative, whereas implicit user feedback can only be positive [Hu et al. 2008]. In explicit user feedback, users can express what they like and don't like. In implicit user feedback, the recommender system can only infer what the users may like. For example, in a music recommender system, the latter may deduce the user preference based on the number of times a track is played. But it is impossible for the recommender system to deduce the reasons for not playing a track—the user may not like the track or may not be aware of the track. Hu et al. [2008] argue that using implicit user feedback without accounting for the missing negative user feedback will misrepresent the user profile.

Range of Users: In explicit user feedback, only a subset of the users of a recommender system expresses user feedback, whereas in implicit user feedback, all users express user feedback. Thus, expressions of implicit user feedback are likely to have less data sparsity than explicit user feedback.

User Transparency: Explicit user feedback is transparent to the user, which has its advantages and disadvantages. The user knows that user feedback may change recommendations, thus providing reasons for providing user feedback. On the other hand, the user may also manipulate the user feedback such that it alters the recommenda- tions [Herlocker et al. 2004]. In contrast, in implicit user feedback, the user may not know which observable behavior leads to recommendations. This makes recommender systems less likely to be manipulated, but at the same time, makes it difficult to explain the recommendations to users.

Bias: Explicit user feedback may be biased toward users who are more expressive than others. The Harper et al. [2005] study of MovieLens users who rate movies found that a disproportionate number of power users contribute ratings. Hence, a recom- mender system based solely on explicit user feedback may be biased toward a particular subset of users.

Importantly, we described earlier that explicit user feedback and implicit user feedback have different properties. But there is no common scale to compare these two forms of user feedback, which gives rise to this open question: which is the better form of user feedback? Intuitively, there is the notion of uncertainty associated with im- plicit user feedback. This is reflected in the paper by Hu et al. [2008], where the view was taken that explicit user feedback refers to user preference, whereas implicit user feedback refers to the confidence in that user preference. On the other hand, Cosley et al. [2003], Hill et al. [1995], and Amatriain et al. [2009a] have shown that explicit user feedback has a degree of uncertainty as well. Other researchers have suggested that in certain systems, implicit user feedback can be more reliable than explicit user feedback [Gadanho and Lhuillier 2007]. Schafer et al. [2007] suggested that implicit user feedback may be more accurate than explicit user feedback in representing the user preference in a music recommender system, although they did not substantiate this claim. Furthermore, when considering the fact that these different properties of explicit and implicit user feedback have a bearing on the performance of the recommender system, a proper design choice between explicit and implicit user feedback is possible only if these two forms of feedback are evaluated within a single framework. Obviously, the inherent assumption in this discussion is that the choice of user feedback in a recommender system is a design choice rather than fixed.

6. TECHNIQUES FOR IMPROVING USER FEEDBACK IN RECOMMENDER SYSTEMS

User feedback constitutes the key input in the user model. Improvements in user feedback imply a better fidelity of the user model and hence better performance of the recommender systems. For example, Amatriain et al. [2009b] have shown that removing noise in 20% of ratings improves the RMSE by more than 5%. In the following sections, we give an overview and comparison of such state-of-the-art techniques. These include reducing the noise in the ratings, using the variance of the ratings, aggregating the ratings from different sources, accounting for the uncertainty in implicit feedback, and combining explicit and implicit user feedback. Note that these techniques are not just about increasing the willingness of users to provide user feedback or the quantity of user feedback available. Instead, they are primarily about increasing the quality of data available.

6.1. Reducing Noise in Rating Data

Noise in a recommender system is divided into natural noise, which refers to noise that the recommender system encounters in the process of collecting, or inferring user preferences and malicious noise, which refers to noise being deliberately inserted into the recommender system [O'Mahony et al. 2006]. In this article, we limit our discussions to natural noise. Explicit and implicit user feedback are inevitably open to natural noise. O'Mahony et al. [2006] detected and eliminated such natural noise in recom- mender systems by defining a threshold *th* within which normalized predicted ratings are allowed to vary from the normalized actual ratings. They carried out experiments to remove noise in the MovieLens and EachMovie datasets, showing that at certain threshold values, discarding those ratings above the threshold value from the recommendation process increased accuracy, albeit at reduced coverage. However, there are two shortcomings to this research. First, experiments lacked a baseline to compare the effect of removal ratings on the accuracy of predictions. A useful baseline could have been randomly removing certain proportions of the ratings. Second, the authors did not explain how they will compute the threshold value *th*. A possible solution could be to minimize a cost function of *th* over the prediction accuracy.

Amatriain et al. [2009b] studied a novel algorithm for removing natural noise in the ratings of movies by re-rating (a process the researchers called *denoising*). In their experiments, they asked users to rate the same movies randomly chosen from the Netflix dataset at three different points in time and different order of movies, hence consti- tuting three datasets. They computed the performance of the three recommendation algorithms (user-based kNN, item-based kNN, and SVD). Their results showed that all three algorithms are affected by natural noise. To denoise the ratings, they com- puted the agreement between ratings from the three datasets. Denoising was carried out in two steps: first, they denoised the ratings from the first dataset using ratings from the second dataset (they called this *one-source re-rating*); second, they denoised the ratings a second time using ratings from the third dataset (called *two-source re- rating*). Accuracy improved by more than 11% by denoising using one-source re-rating and by up to 14.1% using two-source re-rating. They also studied three types of partial denoising of rating data, namely random denoising (improvement of 5% in accuracy over 75% of original ratings using one-source re-rating), data-dependent denoising (removing extreme ratings brought 5% accuracy gain over less than 25% of the original ratings using one-source re-rating), and user- based denoising (removing most inconsistent users brought 5% accuracy gain over 10% of original ratings by 30% of users). User-dependent denoising strategy had the highest performance. In addition, one- source re-rating seems similar in performance to partial denoising. The researchers also calculated that re-rating is a more efficient way of improving the performance of a recommender system than eliciting a new rating. Their results showed that in data-dependent denoising, re-rating contributes 10 times more than a new rating in terms of the improvement of the performance of a recommender system using their denoising algorithm. However, the shortcoming of this solution is that re-rating is not practical. Rating is a very prohibitive process for the user in terms of effort, and thus users have a high inertia to providing ratings. Thus, even with a partial denoising strategy, it may be difficult to get users to re-rate items (i.e., one-source re-rating). Thus, we propose a more practical solution, which is to use implicit user feedback to denoise explicit user feedback.

6.2. Rating Variance

Another technique for improving the performance of recommender systems uses the natural variance in human behavior and activity. There are two forms of variances in ratings, namely user rating variance and item rating variance. User rating vari- ance refers to the variance of ratings given by a user $u \in U$ for a set of items $I_u \subseteq I$, whereas item rating variance refers to the variance of ratings given by a set of users $U_i \subseteq U$ for an item $i \in I$. Amatriain et al. [2009b], discussed in the previous section, deals with user rating variance. Kwon [2008] showed that the accuracy of recommen- dation algorithms decreases as item rating variance increases. Item rating variance affects the recommendation process irrespective of the recommender system algorithm used just as user rating variance, as shown by Amatriain et al. [2009b]. As Kwon [2008] argued, using ratings with small item rating variance will improve the accu- racy of recommender systems. Kwon's novel variation on the standard top-N algorithm [Deshpande and Karypis 2004], which can be abstracted as filtering and ranking processes, is based on three approaches to alter the recommendation set. First, by changing the filtering process (subtracting a portion of the standard deviation of item rating from all item ratings); second, by altering the ranking process (subtracting a portion of standard deviation from the predicted ratings after they were filtered in the standard way); or finally, changing both processes (both filtering and ranking is adjusted by subtract- ing a portion of the standard deviation of item rating). The evaluation was done on metrics of diversity and accuracy [Herlocker et al. 2004]. In empirical results, Kwon [2008] shows that the combined approach had the highest accuracy at the expense of diversity, the ranking approach had the highest diversity, and the filtering approach was more accurate than the ranking approach. Kwon [2008] also showed that the im- pact of the variation on the algorithm depended on the recommendation algorithm used. In addition, Kwon [2008] experimented with the proportion of the standard devi- ation applied to the adjusted ratings and found that accuracy increases and diversity decreases with increasing proportion of the standard deviation. The adjusted filtering approach is the least dependent on the proportion of standard deviation used, whereas the combined approach had the highest dependency.

One shortcoming of this study is that Kwon [2008] did not measure the coverage, making it impossible to objectively analyze the benefit of improved accuracy for the recommender system. Using a similar item rating variance approach, Adomavicius and Kwon [2008] were able to show that they can improve both accuracy and diversity in traditional neighborhood-based collaborative filtering recommender systems.

6.3. Aggregate Rating

Traditional recommender systems estimate unknown ratings from known ratings using techniques such as user-based collaborative filtering [Adomavicius and Tuzhilin 2005]. Umyarov and Tuzhilin [2009] proposed a novel way of enhancing the estimator of an unknown rating by using an external source to provide an optimal linear combination of aggregate average rating and aggregate rating variance, thus called the *aggregate estimator* with the individual estimator. They referred to this as the hierarchical linear regression (HR) model. HR estimates the rating of user u for item i according to Equation (5):

$$r^*_{u,i} = a + \beta \hat{r}_{u,i} \gamma r^a_i, \tag{5}$$

where

$$a = a_0 + a_1 Var(r^a_j),$$
$$\beta = \beta_0 + \beta_1 Var(r^a_j)$$
$$\gamma = \gamma_0 + \gamma_1 Var(r^a_j),$$

$\hat{r}_{u,i}$ is the individual estimator of the rating for user u and item i (provided by traditional RS), and r^a_i is the aggregate estimator of the ratings for item i (provided by an external source). The variables $\hat{r}_{u,i}$, r^a_i, and $Var(r^a_i)$ are known, whereas a_0, a_i, β_0, β_1, γ_0, and γ_1 are unknown and must be estimated from the training sample as described by Umyarov and Tuzhilin [2009]. In fact, as the HR model represents special cases of the classical recommendation algorithm, Umyarov and Tuzhilin [2009] empirically evaluated several cases using the MovieLens dataset and samples of the Netflix dataset to empirically validate their theoretical findings with the mean square error (MSE) accuracy metric. As the external of source of aggregate information, they used the Internet Movie Database [IMDB.com Inc. 1990]. They found that across all of their datasets, the MSE decreased as the number of aggregate ratings added to the recommender system increased. This demonstrated that using aggregate rating in a recommender system can potentially improve its performance. However, their findings cannot be generalized, especially given the surprising result showing that using just the aggregate ratings for prediction outperformed classical recommendation algorithms. This meant that predicting an unknown rating is better done using the mean rating provided by an external source than by using a recommender system. Amatriain et al. [2009b], which we reviewed earlier, showed that classical recommendation algorithms outperformed item average in both intra- and inter-dataset tests. The surprising result obtained by Umyarov and Tuzhilin [2009] may be limited to the movie domain; Herlocker et al. [2004] explained that uniqueness of a dataset can affect the recommender system algorithm. Another interesting finding from the research by Umyarov and Tuzhilin [2009] is that the variance of the aggregate rating did not have an impact as aggregate information. Thus, using only aggregate mean rating was sufficient to improve performance of predicting unknown ratings compared to the traditional recommendation algorithms. Although there are preceding variations on this problem, namely Umyarov and Tuzhilin [2007] and Umyarov and Tuzhilin [2008], the work by Umyarov and Tuzhilin [2009] has presented a generic HR model that is applicable without constraints.

6.4. Uncertainty in Implicit User Feedback

Gadanho and Lhuillier [2007] investigated uncertainly in implicit user feedback in the recommender system domain. In contrast to the work by Amatriain et al. [2009a] on explicit ratings, Gadanho and Lhuillier [2007] did not characterize the uncertainty. They empirically measured the performances of a content-based recommender system algorithm, namely the naïve Bayes classifier. Their evaluations were carried out us- ing implicit and explicit user feedback collected from a recommender system for TV programs. Gadanho and Lhuillier [2007] dealt with the uncertainty of user preference using three approaches. The first approach classified programs that are watched for more than 50% of their duration as matching the user preference and classified them nonmatching otherwise (called *neutral classification*). The second approach follows the same classification rule as the first approach, except it uses the weighting function f (W) $\cos(\frac{w\pi}{100})^2$, where w is the percentage of the program watched, as a means of providing a measure of confidence in the user preference. The final approach is the most brutal, as it classified programs watched for a duration of 90% or higher as matching the user preference and as nonmatching otherwise. A distribution of the percentage of the duration watched showed that those two extremes had the highest proportion of user sessions. However, about half of user sessions fell into the 10% to 90% duration. Thus, this approach classified only the behavior that is the most likely to infer user preference and ignores the ambiguous behaviors. The metrics for evaluation included prediction accuracy, Breese score, precision, and recall [Herlocker et al. 2004]. Gadanho and Lhuillier [2007] evaluated their recommender system by dividing the dataset into 26-week periods, with the recommender system trained iteratively and cumulatively for 1 week, then tested on the following week, with results being averaged over the 26-week period. The results showed similar performances for each approach in terms of accuracy, Breese score, and precision; they were also better than the 'random' rec- ommender system that was used as a baseline. Their results proved inconclusive in terms of determining which is better: the weighting approach or the high-confidence approach. But both were better than the neutral classification. One of the shortcomings of this study is the evaluation; the assumptions used to test the recommendations (i.e., the classification rules and weighting function) were the same ones for inferring user preference. It would have been better to evaluate the recommendations against explicit user preferences. Furthermore, considering that they did not use the coverage metric, the improvement in accuracy cannot objectively be analyzed.

6.5. Combining Implicit User Feedback with Explicit User Feedback

Just like research to unify collaborative and content-based filtering for improved performance [Basilico and Hofmann 2004], unification of explicit and implicit feedback seems a natural step. Gadanho and Lhuillier [2007] and Bell and Koren [2007] have suggested that combining the two forms of feedback would yield the best performance out of recommender systems. In fact, Koren [2010] has shown that using both explicit and implicit user feedback together provides better performance than explicit user feedback on its own. Nevertheless, until recently, there has been a paucity of research examining the unification of the two forms of user feedback into a single framework for modeling user preferences in recommender systems. The key literature on combining explicit and implicit user feedback in the recommender systems include Claypool et al. [2001] (see Section 4), Koren [2010], Liu et al. [2010], Parra and Amatriain [2011], Parra et al. [2011], and Moling et al. [2012].

Koren [2010] has shown that is possible to get better performance compared to solely using explicit user feedback just by using binary implicit user feedback data. For his evaluations, he used the pseudo–implicit user feedback in the Netflix dataset (see Section 2.2). Koren [2010] admitted that such implicit user feedback was not rich but found that nevertheless incorporating this simple implicit user feedback increases the prediction accuracy. Koren [2010] introduced a new neighbor model:

$$\hat{r}_{ui} = b_{ui} + \underset{j\in R(u)}{\quad} (r_{uj} - b_{uj})w_{ij}, \tag{6}$$

where

$\hat{r}_{ui}$is the predicted value of the unknown rating r_{ui},
b_{ui} is the baseline estimate of the unknown rating r_{ui},
w_{ij} is the weight from j to i, representing offsets to the baseline estimates and $r_{uj} - b_{uj}$ viewed as coefficients to these offsets, and
$R(u)$ contains all items for which ratings by u are available.

To add implicit user feedback, the equation is modified as follows:

$$\hat{r}_{ui} = b_{ui} + \underset{j\in R(u)}{(r_{uj} - b_{uj})w_{ij}} + \underset{j\in N(u)}{c_{ij}}, \qquad (7)$$

where

$N(u)$ contains all items for which u has implicit rating, and
c_{ij} is the weight from j to i, representing offsets to the baseline estimate due to the implicit user feedback by u to j.

Koren [2010] expanded this rating prediction model to include item effects, user effects, normalized the sums in the model, and controlled the normalization using coefficients. When evaluated on the Netflix dataset, the model performed better than a model without implicit user feedback. However, the lack of richness of the implicit data limits the generalization of this very interesting result, as richer implicit user feedback data might also have more noise than the binary data used by Koren [2010].

Liu et al. [2010] built a user model of explicit and implicit feedback using the matrix factorization method as used in Hu et al. [2008] and Koren [2010]. They highlighted two issues, namely that the expressions of explicit user feedback and implicit user feedback have different numerical scales and that their accuracies vary significantly. To address these problems, they normalized expressions of both forms of user feedback onto a common scale from 0 to 1 and assigned weights to show the difference in the importance between the two forms of user feedback. Results obtained by Liu et al. [2010] showed that the performance of their model against the baselines depended on the evaluation metrics used and were able to confirm the results obtained by Koren [2010].

Parra and Amatriain [2011] have investigated the relationship between explicit and implicit user feedback. They used implicit user feedback data from Last.fm, namely the number of times an album is played (the playcount) [Herrada 2009b; Jawaheer et al. 2010a]. For the explicit user feedback data, they asked users from which they had the implicit user feedback to rate the albums they listened to. Their objective was to build a model for mapping the implicit user feedback into explicit ratings that could then be used in traditional recommendation algorithms. Parra and Amatriain [2011] found that there is a correlation between implicit user feedback and explicit user feedback, with higher playcounts corresponding to higher ratings and that the recentness of the act of listening to an album affects its ratings, and with albums listened more recently tending to have more positive than negative ratings. To find which was the best model for mapping implicit user feedback (the independent variable) into explicit ratings (the dependent variable), they built several models (such as a model solely using playcount or a model using playcount and recentness. Implicit user feedback (i.e., playcount) was always the common denominator in all models. They found that all models can explain the variance in the data. As for predictive power, they found their results inconclusive, although results showed that all models performed better than the user average baseline with implicit user feedback.

Parra et al. [2011] improved their model of mapping implicit to explicit user feed- back by substituting linear regression with ordinal logistic regression due to the short-comings of linear regression, mainly because linear regression will produce values as

Table III. Comparison of Techniques to Improve User Feedback in Recommender Systems

Publications	User Feedback	Dataset	Improvement Technique	Performance Metric
O'Mahony et al. [2006]	Explicit	EachMovie, MovieLens	Elimination of uncertain ratings beyond threshold value *th*	NMAE overage
Amatriain et al. [2009b]	Explicit	Movie re-rating experiments	Elimination of uncertain ratings using re-rated data	RMSE
Kwon [2008]	Explicit	MovieLens	Adjusting ratings using variance	Diversity, accuracy
Umyarov and Tuzhilin [2009]	Explicit	MovieLens, subsets of Netflix	Adjusting ratings using variance and mean from different data source	MSE, RMSE
Gadanho and Lhuillier [2007]	Implicit	Private dataset on TV programs	Weighted elimination of implicit user feedback	Accuracy, Breese score, precision, recall
Koren [2010]	Explicit and implicit	Netflix	Matrix factorization and new neighborhood model	RMSE
Liu et al. [2010]	Explicit and implicit	Netflix and MovieLens	Matrix factorization with normalized and weighted user feedback	RMSE, NDCG
Parra and Amatriain [2011] and Parra et al. [2011]	Explicit and implicit	Last.fm	Linear and logistic regression models	RMSE, NDCG, MAP
Moling et al. [2012]	Explicit and implicit	Private dataset	Markov decision problem solved using reinforcement learning	Nonstandard evaluation

interval data, whereas explicit ratings are ordinal data. However, Parra et al. [2011] did not provide any comparison data between the two models for mapping implicit user feedback onto explicit user feedback because they used different metrics for evaluating their models. Clearly, further work needs to be done in this area of modeling implicit user feedback as explicit user feedback. Another point about the research by Parra and Amatriain [2011] and Parra et al. [2011] is that there is the possibility of equipment bias in their experiments. They used different interfaces to obtain the explicit and implicit user feedback data (implicit user feedback was obtained from Last.fm, and explicit user feedback was from a user experiment) and in possibly different contexts (e.g., the user experiment was designed solely for eliciting explicit user feedback, and the psyche of the user participating in a user experiment is likely to be different from that when a user is listening to music outside of the lab).

Moling et al. [2012] combined explicit feedback and implicit feedback in an experiment of a client-side recommender system for listening to Internet radio channels (each channel represented a particular genre). For the experiment, the researchers ripped 24 hours of music for each channel (a total of 3,637 music tracks). In their system, ex- plicit feedback is collected through the user specifying which genres of music he likes (expressed in terms of percentage), whereas implicit feedback is collected through the system recording the proportion of the length of the track in the recommended radio channel that the user listened to. A user can only request to change the track that he is listening to, and the system recommends a channel based on his explicit feedback and implicit feedback. The researchers modeled the recommendation task as a Markov decision problem that is solved using reinforcement learning. The system was evaluated by two groups of users (a total of 70 users) experimenting with two solutions. In the first solution, which was the baseline, the system recommended channels solely on probability distribution of explicit genre preferences of the user. In the second solution, the system recommended channels using the explicit genre preferences as well as the implicit feedback (proportion of the track that the user listened to). Statistically significant results showed that there was a 4.76% improvement in the average length of time a user would listen to a track and a 20% improvement in overall use of the system. The researchers also surveyed the participants in the experiment after being exposed to each solution. When questioned whether the system switched the played channel at the right time, the results showed that the improvements were not statistically significant.

6.6. Discussions of the Improvement Techniques Surveyed

In this section, we set up some evaluation criteria and recommender systems' properties to compare the improvement methods and discuss implications for further research.

The metrics used for evaluations such as RMSE, normalized mean absolute error (NMAE), MSE, normalized discounted cumulative gain (NDCG), mean average precision (MAP), coverage, diversity, Breese score, precision, and recall are commonly used in the field. We point readers to Herlocker et al. [2004] for the definitions and references.

The first observation we make is that we are not able to generalize the results of these techniques, as most of them have been evaluated solely on the movie domain. In fact, this criticism can be leveled at most of the published research in the recommender system field. Hence, there is a need for other rich datasets in other domains. On the other hand, using same datasets allows comparative studies. But in this case, we cannot compare the results of the different improvement techniques. Although most of the datasets come from the movie domain, the researchers used different subsets of the dataset [Amatriain et al. 2009b; Umyarov and Tuzhilin 2009], different parameters, or different experimental designs [Amatriain et al. 2009b; Koren 2010].

Another observation is that very often researchers tend to favor accuracy metrics over other metrics. In several cases, the accuracy metrics could not be used to objectively analyze the performance of the recommender system. This criticism applies to papers where performance of the recommender system was due to discarding ratings, and the absence of the coverage metric made it impossible for us to assess the accuracy improvement compared to the utility of the recommender system.

We also note that all evaluations were offline. Hence, these papers could not assess the impact of the improvement techniques on measures like user satisfaction or quality of recommendations.

A final observation is that matrix factorization seems to be the preferred choice of researchers when combining explicit feedback and implicit feedback. Modeling recommender systems as a matrix factorization problem has several benefits, such as superior performance than classic nearest neighbor. Furthermore, matrix factorization model can be extended to cater for implicit feedback, temporal effects, confidence intervals [Koren et al. 2009], and other special characteristics of the data such as item or user bias [Weimer et al. 2008]. Whereas traditional recommendation algorithms struggle with large datasets and sparse explicit feedback, algorithms based on matrix factorization scale linearly and perform well on sparse datasets [Salakhutdinov and Mnih 2008].

Having discussed the different forms of user feedback and the state-of-art-techniques for improving user feedback in recommender systems, in the next section we discuss some of the challenges when employing user feedback in recommender systems.

7. FUTURE CHALLENGES

As stated in the previous section, there are numerous challenges and directions of research to explore in modeling user preferences and better understanding of the user. In this section, we will discuss different abstractions of user feedback, user-centered techniques for improvement of user feedback, the dynamic nature of user preferences, the combination of explicit and implicit user feedback, and the issue of evaluation.

Adomavicius and Tuzhilin [2005] identified and briefly discussed the following challenges for extending the capability of future recommender systems, including understanding users and items, better rating estimations through model-based recommender systems, multidimensional recommender systems, multicriteria ratings, nonintrusiveness of recommender systems, flexibility of recommender systems, and effectiveness of recommender systems. Other challenges that they did not discuss include trust, privacy, scalability, and explainability. All of these are still open research questions. In this section, we will not explore the latter. Instead, we will discuss new challenges that mostly focus on the variability of interactions and behavior of users in recommender systems.

7.1 Abstractions of User Feedback

Herlocker et al. [2004] cite the following reasons that encourage users to provide user feedback: (a) users believe that providing user feedback improves their user profiles and the recommendations, (b) users like to express themselves, (c) users believe that other users will benefit from their contributions, and (d) users believe that they can influence other users' opinions. We think that understanding the reasons users rate items is important in improving how user feedback is employed in recommender systems. An interesting avenue of research is the application of consumer human behavior models to recommender systems. Other interesting approaches could include economic models based on utility theory [Harper et al. 2005] or models based on information theory. Rashid et al. [2008] have applied information theory to the field as a solution to the cold start problem but did not study its application for modeling user feedback. On the other hand, Kluver et al. [2012] used an information theoretic framework to quantify the preference information in ratings and predictions. Their work can be used to compare different rating scales. However, preference is complex, and converting user preference to a numerical value is non-trivial [Lichtenstein and Slovic 2006]. There is substantial literature in psychology on the notion of preference. According to one school of thought, users do not really know what they prefer; instead, they construct their preferences as the situation evolves [Lichtenstein and Slovic 2006].

7.2. User-Centered Improvement of User Feedback

From review of the literature in Section 6, it is clear there is a lack of user-centered investigation of techniques to improve user feedback, such as intelligent user interfaces to encourage user feedback. To illustrate how an intelligent interface can influence performance, we would like to point out the research by Amatriain et al. [2009a], where among other things, better performance was achieved when users were asked to rate items that had similar predicted rating values. Recommender systems could draw from other research adopting user-centered approaches to better understand and model users. For example, profiling users using unified user feedback would provide a personalized experience, as illustrated by Kostkova et al. [2008]. Further, recommender systems can learn from the essential role of qualitative feedback for understanding user preferences, as illustrated in the domain of digital libraries [Kostkova and Madle 2009, 2013].

7.3. Dynamic Nature of User Feedback

The rating interfaces in recommender systems assume that ratings are static. For example, none of the current interfaces for capturing explicit user feedback in recommender systems allows the user to change his or her ratings. But a user's preference is dynamic, temporal, and contextual in nature. In the implicit feedback domain, researchers have shown that taking into account the temporal properties of datasets lead to improvements in rating prediction and ranking [Yang et al. 2012]. Hence, analyzing the contextual nature of user feedback shows promise of performance improvement. This is an area that is being studied in context-aware recommender systems [Ricci et al. 2010; Adomavicius et al. 2011].

7.4. Combining Explicit and Implicit User Feedback into a Single User Model for Recommender Systems

In Section 6.5, we reviewed research that combines explicit and implicit user feedback. There is evidence that performance of recommender systems will improve if we use a combination of explicit and implicit user feedback [Bell and Koren 2007; Koren 2010]. However, more needs to be done in this area. For example, several of the traditional and popular recommendation algorithms, like collaborative filtering using the neighborhood model, have not been adapted for using a combination of explicit and implicit user feedback. Up to now, the algorithms amenable to the combination of explicit and implicit user feedback in recommendations included matrix factorization, linear, and logistic regression [Koren 2010; Parra and Amatriain 2011; Parra et al. 2011; Liu et al. 2010].

7.5. Evaluation

Evaluation of recommender systems is a subject of enormous interest within the research community, as it is still an open question. Herlocker et al. [2004] have extensively reviewed the literature on the evaluation of recommender systems. Offline evaluations are useful for assessing accuracy of recommendations. They have the advantage of providing quick, repeatable, and cheap evaluations on large datasets. However, the lack of richness (implicit user feedback) and diversification in datasets limits the generalization of results and consequent comparisons. In addition, Herlocker et al. [2004] found that in offline evaluations, one can fine-tune the parameters of the recommendation algorithms according to the dataset such that results are artificially positive. Furthermore, offline evaluations cannot assess less tangible attributes of recommendations such as the usefulness, satisfaction, or quality of recommendations [Herlocker et al. 2004]. The alternative to offline evaluation is conducting live user experiments [Herlocker et al. 2004] for a more holistic approach to evaluation where those less tangible yet important features are assessed. However, live evaluations of recommender systems, although highly desirable, are rare, as they are costly and lengthy, and so are virtually nonexistent. This is an important direction for future research. However, it is paramount to draw lessons from body of research from other domains experimenting with evaluation with real-world users and systems and additional challenges identified for collecting user feedback First, users often perceive their online behaviors and preferences differently from their actual behaviors [Madle et al. 2009; Roy et al. 2010]. Additionally, live user experiments, especially in the case of a prototype or research project, often suffer from unpolished user interfaces and the propensity of bugs, which affects the user experience and hence the outcome of the evaluation [Oliver et al. 2009].

It is even more challenging to evaluate recommender systems that process implicit user feedback. For example, in contrast to explicit user feedback, where RMSE is quite commonly used in the literature, there is no standard metric for measuring the performance of recommender systems processing implicit user feedback. This makes it difficult to compare the performances of such recommender systems. Given the seamless collection of implicit user feedback, there is a need to research the evaluation of recommender systems that process implicit user feedback.

8. CONCLUSION

In this article, we presented a classification framework for comparing explicit and implicit user feedback in recommender systems based on a set of distinct properties. These include cognitive effort, user model, scale of measurement, domain relevance, sensitivity to noise, polarity, range of users, user transparency, and bias. This enabled us to classify recommender systems utilizing user feedback and highlight projects that addressed the use of explicit and implicit user feedback in recommender systems as key mechanisms for modeling user preferences in items. We identified avenues for future research, including the need to combine these two forms of feedback in a single framework, as they are not readily comparable at this time. The limited approaches and datasets currently available are subject to noise, uncertainty, and human natural variance in preferences. But little research in recommender systems has accounted for these. Fur- thermore, we outlined how improvements in recommender systems performance can be achieved more efficiently. In particular, we identified several research challenges lying ahead, such as a unified user model combining explicit and implicit user feedback in the same user model. Further, we also identified that, more importantly, novel ap- proaches are required to assess less tangible features about recommendations through live and holistic evaluations. This article highlights the challenges for future research with focus given to key aspects such as the user model for recommender systems.

REFERENCES

Silvana Aciar, Debbie Zhang, Simeon Simoff, and John Debenham. 2006. Recommender system based on consumer product reviews. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI'06)*. IEEE Computer Society, Washington, DC, 719–723.

Gediminas Adomavicius and YoungOk Kwon. 2008. Overcoming accuracy-diversity tradeoff in recommender systems: A variance-based approach. In *Proceedings of the 18th Workshop on Information and Technology and Systems (WITS'08)*.

Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alex Tuzhilin. 2011. Context-aware recommender systems. *AI Magazine* 32, 3, 67–80.

Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6, 734–749. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1423975.

Xavier Amatriain. 2011. Recommender Systems: We're Doing It (All) Wrong. Retrieved May 26, 2014, from http://technocalifornia.blogspot.com/2011/04/recommender-systems-were-doing-it-all.html.

Xavier Amatriain, Josep Pujol, and Nuria Oliver. 2009a. I like it. I like it not: Evaluating user ratings noise in recommender systems. In G.-J. Houben, G. McCalla, F. Pianesi, and M. Zancanaro (Eds.), *User Modeling, Adaptation, and Personalization*. Springer, Berlin, Heidelberg, 247–258.

Xavier Amatriain, Josep M. Pujol, Nava Tintarev, and Nuria Oliver. 2009b. Rate it again: Increasing recommendation accuracy by user re-rating. In *Proceedings of the 3rd Conference on Recommender Systems*. ACM Press, New York, NY, 173–180.

Justin Basilico and Thomas Hofmann. 2004. Unifying collaborative and content-based filtering. In *Pro- ceedings of the 21st International Conference on Machine Learning (ICML'04)*. ACM Press, New York, NY, 9.

Robert M. Bell and Yehuda Koren. 2007. Lessons from the Netflix prize challenge. *ACM SIGKDD Explo- rations Newsletter* 9, 2, 75. http://portal.acm.org/citation.cfm?doid=1345448.1345465.

James Bennett and Stan Lanning. 2007. The Netflix prize. In *Proceedings of the KDD-Cup and Workshop at the 13th SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Mark Claypool, Phong Le, Makoto Wased, and David Brown. 2001. Implicit interest indicators. In *Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI'01)*. ACM Press, New York, NY, 33–40.

Dan Cosley, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. 2003. Is seeing believ- ing? How recommender interfaces affect users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'03)*. ACM Press, New York, NY, 585–592. http://portal.acm.org/citation.cfm?doid=642611.642713.

James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The YouTube video recommendation system. In *Proceedings of the 4th Conference on Recommender Systems*. ACM Press, New York, NY, 293–296.

Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM Transactions on Information Systems* 22, 1, 143–177. http://portal.acm.org/citation.cfm?doid=963770.963776.

Christian Desrosiers and George Karypis. 2010. A novel approach to compute similarities and its application to item recommendation. In *Proceedings of the 11th Pacific Rim International Conference on Trends in Artificial Intelligence*. Springer-Verlag, Berlin, Heidelberg, 39–51.

Andy Field and Graham Hole. 2003. *How to Design and Report Experiments*. Sage Publications.

Sandra C. Gadanho and Nicolas Lhuillier. 2007. Addressing uncertainty in implicit preferences. In *Proceedings of the 2007 Conference on Recommender Systems*. ACM Press, New York, NY, 97–104.

Sandra Garcia Esparza, Michael P. O'Mahony, and Barry Smyth. 2012. Mining the real-time Web: A novel approach to product recommendation. *Journal of Knowledge-Based Systems: Special Issue on Innovative Techniques and Applications of Artificial Intelligence* 29, 3–11.

Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval* 4, 2, 133–151. http://www.springerlink.com/index/M5458KV8LJ602646.pdf.

GroupLens Research. 2011a. EachMovie Dataset. Retrieved May 26, 2014, from http://www.grouplens.org/node/76.

GroupLens Research. 2011b. MovieLens Dataset. Retrieved May 26, 2014, from http://www.grouplens.org/node/73.

Maxwell F. Harper, Xin Li, Yan Chen, and Joseph A. Konstan. 2005. An economic model of user rating in an online recommender system. In L. Ardissono, P. Brna, and A. Mitrovic (Eds.), *User Modeling 2005*. Springer, 307–316.

Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22, 1, 5–53. http://portal.acm.org/citation.cfm?doid=963770.963772.

Oscar Celma Herrada, 2009a. Last.fm Dataset. Retrieved May 26, 2014, from http://www.dtic.upf.edu/∼ocelma/MusicRecommendationDataset/index.html.

Oscar Celma Herrada. 2009b. *Music Recommendation and Discovery in the Long Tail.* Ph.D. Dissertation.

Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. 1995. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*. 194–201. http://portal.acm.org/citation.cfm?doid=223904.223929.

Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 8th IEEE International Conference on Data Mining*. 263–272. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4781121.

Zan Huang. 2007. Selectively acquiring ratings for product recommendation. In *Proceedings of the 9th International Conference on Electronic Commerce*. ACM Press, New York, NY, 379–388.

IMDB.com Inc. 1990. The Internet Movie Database. Retrieved May 26, 2014, from http://www.imdb.com.

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Friedrich Gerhard. 2011. *Recommender Systems: An Introduction*. Cambridge University Press.

Gawesh Jawaheer, Martin Szomszor, and Patty Kostkova. 2010a. Characterisation of explicit feedback in an online music recommendation service. In *Proceedings of the 4th Conference on Recommender Systems*. ACM Press, New York, NY, 317–320.

Gawesh Jawaheer, Martin Szomszor, and Patty Kostkova. 2010b. Comparison of implicit and explicit feedback from an online music recommendation service. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*. ACM Press, New York, NY, 47–51.

Jester Dataset. 2012. Anonymous Ratings from the Jester Online Joke Recommender System. Retrieved May 26, 2014, from http://eigentaste.berkeley.edu/jester-data/.

Jinmook Kim and Douglas W. Oard. 2001. User modeling for information access based on implicit feedback. In *Proceedings of the Symposium of ISKO-France*. 1–11.

Daniel Kluver, Tien T. Nguyen, Michael Ekstrand, Shilad Sen, and John Riedl. 2012. How many bits per rating? In *Proceedings of the 6th Conference on Recommender Systems*. ACM Press, New York, NY, 99–206.

Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. 1997. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM* 40, 3, 87. http://portal.acm.org/citation.cfm?id=245108.245126.

Yehuda Koren. 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data* 4, 1, 1–24. http://portal.acm.org/citation.cfm?doid=1644873.1644874.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8, 30–37.

Yehuda Koren and Joe Sill. 2011. OrdRec: An ordinal model for predicting personalized item rating distributions. In *Proceedings of the 5th Conference on Recommender Systems (RecSys'11)*. ACM Press, New York, NY, 117–124.

Patty Kostkova, Gayo Diallo, and Gawesh Jawaheer. 2008. User profiling for semantic browsing in medical digital libraries. In *The Semantic Web: Research and Applications*. Springer, 827–831.

Patty Kostkova and Gemma Madle. 2009. User-centered evaluation model for medical digital libraries. In *Knowledge Management for Health Care Procedures*. Springer, 92–103.

Patty Kostkova and Gemma Madle. 2013. What impact do healthcare digital libraries have? An evaluation of national resource of infection control at the point of care using the Impact-ED framework. *International Journal on Digital Libraries* 13, 2, 77–90.

YoungOk Kwon. 2008. Improving top-N recommendation techniques using rating variance. In *Proceedings of the 2008 Conference on Recommender Systems (RecSys'08)*. ACM Press, New York, NY, 307–310.

Juha Leino and Kari-Jouko Raiha. 2007. Case amazon: Ratings and reviews as part of recommendations. In *Proceedings of the 2007 Conference on Recommender Systems (RecSys'07)*. ACM Press, New York, NY, 137–140.

Sarah Lichtenstein and Paul Slovic (Eds.). 2006. *The Construction of Preference*. Cambridge University Press.

Nathan N. Liu, Evan W. Xiang, Min Zhao, and Qiang Yang. 2010. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th Conference on Information and Knowledge Management (CIKM'10)*. ACM Press, New York, NY, 1445–1448.

Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*. ACM Press, New York, NY, 131–140.

Gemma Madle, Anouk Berger, Sebastien Cognat, Sylvio Menna, and Patty Kostkova. 2009. User information seeking behaviour: Perceptions and reality. An evaluation of the WHO Labresources Internet portal. *Informatics for Health and Social Care* 34, 1, 30–38.

Benjamin M. Marlin and Richard S. Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys'09)*. ACM Press, New York, NY, 5–12. http://portal.acm.org/citation.cfm?doid=1639714.1639717.

Lorraine McGinty and Barry Smyth. 2002. Evaluating preference-based feedback in recommender systems. In *Artificial Intelligence and Cognitive Science*. Springer, 209–214.

Lorraine McGinty and Barry Smyth. 2005. *Improving the Performance of Recommender Systems That Use Critiquing*. Springer.

Omar Moling, Linas Baltrunas, and Francesco Ricci. 2012. Optimal radio channel recommendations with explicit and implicit feedback. In *Proceedings of the 6th ACM Conference on Recommender Systems (RecSys'12)*. ACM Press, New York, NY, 75–82.

David M. Nichols. 1997. Implicit rating and filtering. In *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*.

Douglas W. Oard. 1998. Implicit feedback for recommender systems. In *Proceedings of the AAAI Workshop on Recommender Systems*.

Douglas W. Oard and Jinmook Kim. 2001. Modeling information content using observable behavior. In *Proceedings of the 64th Annual Conference of the American Society for Information Science and Technology*.

Helen Oliver, Gayo Diallo, Ed De Quincey, Dimitra Alexopoulou, Bianca, Habermann, Patty Kostkova, Michael Schroeder, Simon Jupp, Khaled Khelif, Robert Stevens, Gawesh Jawaheer, and Madle Gawesh. 2009. A user-centered evaluation framework for the Sealife semantic Web browsers. *BMC Bioinformatics* 10, Suppl. 10, 1.

Michael P. O'Mahony, Neil J. Hurley, and Guénolé C. M. Silvestre. 2006. Detecting noise in recommender system databases. In *Proceedings of the 11th International Conference on Intelligent User Interfaces (IUI'06)*. ACM Press, New York, NY, 109. http://portal.acm.org/citation.cfm?doid=1111449.1111477.

.

Denis Parra and Xavier Amatriain. 2011. Walk the talk: Analyzing the relation between implicit and explicit feedback for preference elicitation. In *Proceedings of the 19th International Conference on User Modeling, Adaptation, and Personalization (UMAP'11)*. Springer-Verlag, Berlin, Heidelberg, 255–268.

Denis Parra, Alexandros Karatzoglou, Xavier Amatriain, and Idil Yavuz. 2011. Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. In *Proceedings of CARS 2011*.

Al Manunur Rashid, George Karypis, and John Riedl. 2008. Learning preferences of new users in recommender systems: An information theoretic approach. *ACM SIGKDD Explorations Newsletter* 10, 2, 90–100. http://dl.acm.org/citation.cfm?id=1540302.

Steffen Rendle, Christoph Freudenthaler, Gantner Zeno, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, Arlington, VA, 452–461.

Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). 2010. *Recommender Systems Handbook*. Springer. http://www.springerlink.com/index/10.1007/978-0-387-85820-3.

Judy Robertson. 2011. Stats: We're Doing It Wrong. Retrieved May 26, 2014, from http://cacm.acm.org/blogs/blog-cacm/107125-stats-were-doing-it-wrong/fulltext.

Anjana Roy, Patty Kostkova, Mike Catchpole, and Ewart Carson. 2010. "Do users do what they think they do?" A comparative study of user perceived and actual information searching behaviour in the National Electronic Library of Infection. In *Electronic Healthcare*. Springer, 96–103.

Ruslan Salakhutdinov and Andriy Mnih. 2008. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*. Vol. 20, 1257–1264.

J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.), *The Adaptive Web*. Springer, Berlin, Heidelberg, 291–324.

Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. 2010. How useful are your comments?: Analyzing and predicting YouTube comments and comment ratings. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. ACM Press, New York, NY, 891–900.

E. Isaac Sparling and Shilad Sen. 2011. Rating: How difficult is it? In *Proceedings of the 5th Conference on Recommender Systems*. ACM Press, New York, NY, 149–156.

Martin Szomszor, Ciro Cattuto, Harith Alani, Kieron O'Hara, Andrea Baldassarri, Vittorio Loreto, and Vito D. P. Servedio. 2007. Folksonomies, the semantic Web, and movie recommendation. In *Proceedings of the 4th European Semantic Web Conference, Bridging the Gap Between Semantic Web and Web 2.0*.

Akhmed Umyarov and Alexander Tuzhilin. 2007. Leveraging aggregate ratings for better recommendations. In *Proceedings of the 2007 Conference on Recommender Systems (RecSys'07)*. ACM Press, New York, NY, 161. http://portal.acm.org/citation.cfm?doid=1297231.1297261.

Akhmed Umyarov and Alexander Tuzhilin. 2008. Improving collaborative filtering recommendations using external data. In *Proceedings of the 2008 8th International Conference on Data Mining*. IEEE Computer Society, Washington, DC, 618–627. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber = 4781157.

Akhmed Umyarov and Alexander Tuzhilin. 2009. Improving rating estimation in recommender systems using aggregation- and variance-based hierarchical models. In *Proceedings of the 3rd Conference on Recommender Systems (RecSys'09)*. ACM Press, New York, NY, 37. http://portal.acm.org/citation.cfm?doid=1639714.1639722.

Markus Weimer, Alexandros Karatzoglou, and Alex Smola. 2008. Improving maximum margin matrix factorization. *Machine Learning* 72, 3, 263–276.

Diyi Yang, Tianqi Chen, Weinan Zhang, Qiuxia Lu, and Yong Yu. 2012. Local implicit mining for music recommendation. In *Proceedings of the 5th Conference on Recommender Systems*. ACM Press, New York, NY, 91–98.

Markus Zanker and Markus Jessenitschnig. 2009. Case-studies on exploiting explicit customer requirements in recommender systems. *User Modeling and User-Adapted Interaction* 19, 1, 133–166.

Cai-Nicolas Ziegler. 2004. Book-Crossing Dataset. Retrieved May 26, 2014, from http://www.informatik.uni-freiburg.de/∼cziegler/BX/.