On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected

PANAGIOTIS ADAMOPOULOS and ALEXANDER TUZHILIN, New York University

Although the broad social and business success of recommender systems has been achieved across several domains, there is still a long way to go in terms of user satisfaction. One of the key dimensions for significant improvement is the concept of *unexpectedness*. In this article, we propose a method to improve user satisfaction by generating unexpected recommendations based on the utility theory of economics. In particular, we propose a new concept of unexpectedness as recommending to users those items that depart from what they would expect from the system - the consideration set of each user. We define and formalize the concept of unexpectedness and discuss how it differs from the related notions of novelty, serendipity, and diversity. In addition, we suggest several mechanisms for specifying the users' expectations and propose specific performance metrics to measure the unexpectedness of recommendation lists. We also take into consideration the quality of recommendations using certain utility functions and present an algorithm for providing users with unexpected recommendations of high quality that are hard to discover but fairly match their interests. Finally, we conduct several experiments on "real-world" datasets and compare our recommendation results with other methods. The proposed approach outperforms these baseline methods in terms of unexpectedness and other important metrics, such as coverage, aggregate diversity and dispersion, while avoiding any accuracy loss.

Categories and Subject Descriptors: H.1.2 [**Models and Principles**]: User/Machine Systems—*Human information processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering, Selection process*; H.4.m [**Information Systems Applications**]: Miscellaneous

General Terms: Algorithms, Design, Experimentation, Human Factors, Measurement, Performance, Theory

Additional Key Words and Phrases: Diversity, evaluation, novelty, recommendations, recommender systems, serendipity, unexpectedness, utility theory

ACM Reference Format:
Panagiotis Adamopoulos and Alexander Tuzhilin. 2014. On unexpectedness in recommender systems: Or how to better expect the unexpected. ACM Trans. Intell. Syst. Technol. 5, 4, Article 54 (December 2014), 32 pages.
DOI: http://dx.doi.org/10.1145/2559952

> "If you do not expect it, you will not find the unexpected, for it is hard to find and difficult."
>
> – Heraclitus of Ephesus, 544-484 B.C.

This paper is based on preliminary research [Adamopoulos and Tuzhilin 2011] presented in the Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011), at the 5th ACM International Conference on Recommender Systems (RecSys 2011).

Authors' addresses: P. Adamopoulos and A. Tuzhilin, Department of Information, Operations & Management Sciences (IOMS), Leonard N. Stern School of Business, New York University; email: {padamopo, atuzhili}@stern.nyu.edu.

DOI: http://dx.doi.org/10.1145/2559952

1. INTRODUCTION

Over the past decade, a wide variety of different types of recommender systems (RSes) has been developed and used across several domains [Adomavicius and Tuzhilin 2005]. Although the broad social and business acceptance of RSes has been achieved and the recommendations of the latest class of systems are significantly more accurate than they used to be a decade ago [Bell et al. 2009], there is still a long way to go in terms of satisfaction of users' actual needs [Konstan and Riedl 2012]. This is primarily due to the fact that many existing RSes focus on providing more accurate rather than more novel, serendipitous, diverse, and useful recommendations. Some of the main problems pertaining to this narrow accuracy-based focus of many existing RSes [Cremonesi et al. 2011; Adamopoulos 2013a; Adamopoulos and Tuzhilin 2014] and ways to broaden the current approaches have been discussed in McNee et al. [2006] and Adamopoulos [2014]. One key dimension for improvement that can significantly contribute to the overall performance and usefulness of RSes, which is still underexplored, is the notion of *unexpectedness*. RSes often recommend expected items with which users are already familiar and thus are of little interest to them. For example, a shopping RS may recommend to customers products such as milk and bread. Although accurate, in the sense that the customer will indeed buy these two products, such recommendations are of little interest because they are obvious, since the shopper will most likely buy these products even without these recommendations. Therefore, because of this potential for higher user satisfaction, it is important to study non-obvious recommendations. Motivated by the challenges and implications of this problem, we try to resolve it by recommending unexpected items of significant usefulness to users. Following the Greek philosopher Heraclitus, we approach this hard and difficult problem of finding and recommending unexpected items by first capturing the expectations of the user. The challenge is not only to identify the items expected by the users and then derive unexpected ones, but also to enhance the concept of unexpectedness while still delivering recommendations of high quality that achieve a fair match to users' interests. In this article, we formalize this concept by providing a new formal definition of unexpected recommendations as those recommendations that significantly depart from each user's expectations and consideration set, and we differentiate it from various related concepts, such as novelty and serendipity. We also propose a novel method for generating unexpected recommendations and suggest specific metrics to measure the unexpectedness of recommendation lists. Finally, we show that the proposed method can enhance unexpectedness while maintaining the same or higher levels of accuracy of recommendations.

2. RELATED WORK AND CONCEPTS

In the past, some researchers tried to provide alternative definitions of unexpectedness and various related but still different concepts, such as novelty, diversity, and serendipity. In the following sections, we discuss the aforementioned concepts and how they differ from the proposed notion of unexpectedness.

2.1. Novelty

In particular, *novel* recommendations are recommendations of new items that the user did not know about [Konstan et al. 2006]. Among the works that attempt to increase the novelty of recommendations, Hijikata et al. [2009] use Collaborative Filtering (CF) and explicitly ask users what items they already know. In addition, Weng et al. [2007] suggest a taxonomy-based RS that utilizes hot-topic detection using association rules to improve novelty and quality of recommendations, whereas Zhang and Hurley [2009] propose to enhance novelty at a small cost to overall accuracy by partitioning the user profile into clusters of similar items and then recommending lists of items that match well with each cluster, rather than with the entire user profile. Also, Celma and Herrera [2008] analyze the item-based recommendation network to detect whether its intrinsic topology has a pathology that hinders "long-tail" novel recommendations. Finally, Nakatsuji et al. [2010] define and measure novelty as the smallest distance from the class the user accessed previously to the class that includes the target item.

However, comparing novelty to unexpectedness, a novel recommendation might be unexpected, but novelty is strictly defined in terms of previously unknown, non-redundant items, without allowing for known but unexpected ones. Also, novelty does not include any positive reactions of the user to recommendations. Illustrating some of these differences in the movie context, assume that the user John Doe is mainly interested in Action & Adventure films. Recommending to this user the newly released production of one of his favorite Action & Adventure film directors is a novel recommendation, but not necessarily unexpected and possibly of low utility for him since John was either expecting the release of this film or he could easily find out about it. Similarly, assume that we recommend to this user the latest Children & Family film. Although this is definitely a novel recommendation, it is probably also of low utility and would be likely considered "irrelevant" because it departs too much from his expectations.

2.2. Serendipity

Serendipity, the most closely related concept to unexpectedness, involves a positive emotional response of the user about a previously unknown (novel) item and measures how surprising these recommendations are [Shani and Gunawardana 2011]. Serendipitous recommendations are, by definition, also novel. However, a serendipitous recommendation involves an item that the user would not be likely to discover otherwise, whereas the user might autonomously discover novel items. Working on serendipity, Iaquinta et al. [2008] propose to recommend items whose description is semantically far from users' profiles and Kawamae et al. [2009] and Kawamae [2010] suggest a recommendation algorithm based on the assumption that users follow earlier adopters who have demonstrated similar preferences. In addition, Sugiyama and Kan [2011] propose a method for recommending scholarly papers utilizing dissimilar users and co-authors to construct the profile of the target researcher. Also, André et al. [2009] examine the potential for serendipity in Web search and suggest that information about personal interests and behavior may be used to support serendipity.

Nevertheless, even though both serendipity and unexpectedness involve a positive surprise for the user, serendipity is restricted to novel items and their accidental discovery, without taking into consideration the expectations of the users and the relevance of the items, and thus it constitutes a different type of recommendation that can be more risky and ambiguous. To further illustrate the differences between these two concepts, let us assume that we recommend to John Doe the latest Romance film. There are some chances that John will like this novel item and the accidental discovery of a serendipitous recommendation. However, such a recommendation might also be of low utility to the user since it does not take into consideration his expectations and the relevance of the items. On the other hand, assume that we recommend to John Doe a movie in which one of his favorite Action & Adventure film directors is performing as an actor in an old (not novel) Action film of another director. The user will most probably like this unexpected but not serendipitous recommendation.

2.3. Diversity

Diversification is defined as the process of maximizing the variety of items in a recommendation list. Most of the literature in Recommender Systems and Information Retrieval studies the principle of diversity to improve user satisfaction. Typical approaches replace items in the derived recommendation lists in order to minimize similarity between all items or simply remove “obvious” items from them [Billsus and Pazzani 2000]. For instance, Zhang and Hurley [2008] focus on intralist diversity to address the problem as the joint optimization of two objective functions reflecting preference similarity and item diversity. In addition, Zhang et al. [2012] propose a collection of algorithms to simultaneously increase novelty, diversity, and serendipity at low cost to accuracy, and Zhou et al. [2010] suggest a hybrid algorithm that, without relying on any semantic or context-specific information, simultaneously gains in both accuracy and diversity of recommendations. In addition, Said et al. [2012] suggest an inverted nearest neighbor model and recommend items disliked by the least similar users, whereas Adamopoulos and Tuzhilin [2013a] propose a probabilistic neighborhood selection method that also improves both diversity and accuracy of recommendations by selecting a diverse but representative set of neighbors. Following a different direction, McSherry [2002] investigates the conditions in which similarity can be increased without loss of diversity and presents an approach to retrieval that is designed to deliver such similarity-preserving increases in diversity. In other research streams, Panniello et al. [2009] compare several contextual pre-filtering, post-filtering, and contextual modeling methods in terms of accuracy and diversity of their recommendations to determine which methods outperform others and under which circumstances. Considering how to measure diversity, Castells et al. [2011] and Vargas and Castells [2011] aim to cover and generalize the metrics reported in the recommender systems literature [Zhang and Hurley 2008; Zhou et al. 2010; Ziegler et al. 2005] and derive new ones by taking into consideration item position and relevance.

Examining similar but yet different concepts of diversity, Ziegler et al. [2005] propose a similarity metric using a taxonomy-based classification and use this to assess the topical diversity of recommendation lists. They also provide a heuristic algorithm to increase the diversity of the recommendation lists. Then, Adomavicius and Kwon [2009, 2012] propose the concept of aggregate diversity as the ability of a system to recommend across all users as many different items as possible while keeping accuracy loss to a minimum by a controlled promotion of less popular items toward the top of the recommendation lists. Finally, Lathia et al. [2010] consider the concept of temporal diversity (i.e., diversity in the sequence of recommendation lists produced over time).

Taking into consideration the different notions and concepts discussed so far, avoiding a too narrow set of choices is generally a good approach to increase the usefulness of a recommendation list since it enhances the chances that a user is pleased by at least some recommended items. However, diversity is a very different concept from unexpectedness and constitutes an ex-post process that can be combined with the concept of unexpectedness.

2.4. Unexpectedness

Pertaining to *unexpectedness*, in the field of knowledge discovery, several authors [Silberschatz and Tuzhilin 1996; Berger and Tuzhilin 1998; Padmanabhan and Tuzhilin 1998, 2000, 2006] propose a characterization relative to the system of prior domain beliefs and develop efficient algorithms for the discovery of unexpected patterns that combine the independent concepts of unexpectedness and minimality of patterns. Kontonasios et al. [2012] survey different methods for assessing the unexpectedness of patterns focusing on frequent itemsets, tiles, association rules, and classification rules. In the field of recommender systems, Murakami et al. [2008] and Ge et al. [2010] both suggest a definition of unexpectedness as the difference in predictions between two algorithms, the deviation of a recommender system from the results obtained from a primitive prediction model that shows high ratability, and corresponding metrics for evaluating this system-centric notion of unexpectedness. In addition, Akiyama et al.

[2010] propose unexpectedness as a general metric that does not depend on a user's record and only involves an unlikely combination of item features.

However, all these system-centric and item-based approaches do not fully capture the multifaceted concept of unexpectedness since they do not truly take into account the actual *expectations of the users*, which is crucial according to philosophers such as Heraclitus and some modern researchers [Silberschatz and Tuzhilin 1996; Berger and Tuzhilin 1998; Padmanabhan and Tuzhilin 1998]. Hence, an alternative user-centric definition of unexpectedness that takes into account prior expectations of the users as well as novel methods for providing to users unexpected recommendations are still needed. In particular, a user-centric definition of unexpectedness and the corresponding methods should avoid recommendations that are obvious, irrelevant, or expected by the user, but without being strictly restricted only to novel items, and at the same time it should also allow for a notion of pleasant discovery since a recommendation makes more sense when it exposes the user to a relevant experience that she or he has not thought of or experienced yet. In this article, we deviate from the previous definitions of unexpectedness and propose a new formal user-centric definition that recommends to users those items that are not included in their consideration sets and depart from what they would expect from the recommender system.

3. DEFINITION OF UNEXPECTEDNESS

In this section, we formally model and define the concept of unexpected recommendations as those recommendations that significantly depart from the user's expectations. However, unexpectedness alone is not enough to provide truly useful recommendations since it is possible to deliver unexpected recommendations but of low quality. Therefore, after defining unexpectedness, we introduce the idea of recommendation *utility* and provide an example of utility as a function of the *quality* of a recommendation (e.g., specified by the item's rating) *and* its *unexpectedness*. We maintain that this utility of a recommended item is the concept on which we should focus (vis-à-vis "pure" unexpectedness) by recommending items with the highest levels of utility to the user. Finally, we propose an algorithm for providing users with unexpected recommendations of high quality that are hard to discover but fairly match their interests, and then present specific performance measures for evaluating the unexpectedness of the generated recommendations. We define unexpectedness in Section 3.1, we model the utility of recommendations in Section 3.2, and we propose a method for delivering unexpected recommendations of high quality in Section 3.3 and metrics for their evaluation in Section 3.4.

3.1. Unexpectedness

To define unexpectedness, we first start with the user expectations. The *expected items* for each user u can be defined as a *consideration set*, a finite collection of typical items that the user considers as choice candidates to serve her own current needs or fulfill her intentions, as indicated by her interactions with the recommender system. This concept of the set of user expectations can be more precisely specified and operationalized in the lower level of a specific application and recommendation setting. In particular, the set of expected items E_u for a user can be specified in various ways, such as the set of past transactions performed by the user or as a set of "typical" recommendations that she or he expects to receive or has received in the past. Moreover, the sets of user expectations, as the true expectations of the users, can also be adapted to different contexts and evolve over time. For example, in the case of a movie RS, this set of expected items may include all the movies already seen by the user *and* all the related and similar movies, where "relatedness" and "similarity" are specified and operationalized through specific mechanisms described in Section 4. Intuitively, an item included in the set of expected recommendations derives "zero unexpectedness" for the user, whereas the more an item departs from the set of expectations, the more unexpected it is, until it starts being perceived as irrelevant by the user. Unexpectedness should thus be a positive, unbounded function of the distance of this item from the set of expected items. More formally, we define *unexpectedness* in recommender systems as follows. First, we define:

$$\delta_{u,i} = d(i; \mathrm{E}_u), \tag{1}$$

where $d(i; \mathrm{E}_u)$ is the distance of item i from the set of expected items E_u for user u. Then, taking into consideration the relevance of the item, *utility of unexpectedness* of item i with respect to user expectations E_u is defined as some unimodal function Δ of this distance:

$$\Delta(\delta_{u,i}; \delta^*_u), \tag{2}$$

where δ^*_u is the best (most preferred) unexpected distance from the set of expected items E_u for user u (the mode of distribution Δ). In particular, the most preferred unexpected distance δ^*_u for user u is a horizontally differentiated feature and can be interpreted as the distance that results in the highest utility for a given quality of an item (see Section 3.2) and captures the preferences of the user about unexpectedness. Intuitively, unimodality of this function Δ indicates that:

(1) there is only one *most preferred unexpected* distance;
(2) an item that greatly departs from user's expectations, even if it results in a large departure from expectations, will be probably perceived as irrelevant by the user and, hence, it is not truly unexpected; and
(3) items that are close to the expected set are not truly unexpected but rather obvious to the user.

These definitions[1] clearly take into consideration the actual *expectations of users* as we discussed in Section 2. Hence, unexpectedness is neither a characteristic of items nor users, since an item can be expected for a specific user but unexpected for another. It is the interplay of the user and the item that characterizes whether the particular recommendation is unexpected for the specific user. However, recommending to a user those items that result in the highest level of unexpectedness could be problematic, since recommendations should also be of high quality and fairly match user preferences. In other words, it is important to emphasize that simply increasing the unexpectedness of a recommendation list is valueless if this list does not contain relevant items of high quality that the user likes. To generate recommendations that would maximize the users' satisfaction, we use certain concepts from utility theory in economics [Marshall 1920].

3.2. Utility of Recommendations

Pertaining to unexpectedness recommender systems, while trying to keep the complexity of our method to a minimum, we specify the utility of a recommendation of an item to a user in terms of two components: the utility of quality that the user will gain from the recommended item and the utility of unexpectedness of this item, as defined in Section 3.1. The proposed model follows the standard assumption in economics that users are engaging in optimal utility maximizing behavior [Marshall 1920]. Additionally, we consider the quality of an item to be a vertically differentiated characteristic [Tirole 1988], which means that utility is a monotone function of quality; hence, given the unexpectedness of an item, the greater the quality of this item, the greater the

[1] The aforementioned definitions serve as templates for the proposed concepts that are precisely defined in Sections 4.2.1–4.2.4. Unless otherwise stated, the terms "unexpectedness" and "utility of unexpectedness" are used interchangeably.

utility of the recommendation to the user. Consequently, without loss of generality, we can estimate this overall utility of a recommendation using the previously mentioned utility of quality and the loss in utility by the departure from the preferred level of unexpectedness δ_u^*. This allows the utility function to have the required characteristics described so far. Note that utility as a function of unexpectedness and quality is nonlinear, bounded, and experiences a global maximum. Formalizing these concepts to provide an example of a utility function to illustrate the proposed method, we assume that each user u values the quality of an item by a positive constant q_u and that the quality of item i is represented by the corresponding rating $r_{u,i}$. Then, we define the utility derived from the quality of the recommended item i to the user u as:

$$U_{u,i}^q = q_u \times r_{u,i} + \epsilon_{u,i}^q, \tag{3}$$

where $\epsilon_{u,i}^q$ is the error term defined as a random variable capturing the stochastic aspect of recommending item i to user u. We also assume that user u values the unexpectedness of an item by a non-negative factor λ_u measuring the user's tolerance to redundancy and irrelevance. The utility to the user decreases by departing from the preferred level of unexpectedness δ_u^*. Then, the loss of utility caused by departing from the preferred level of unexpectedness of a recommendation can be represented as:

$$U_{u,i}^\delta = -\lambda_u \times \phi(\delta_{u,i}; \delta_u^*) + \epsilon_{u,i}^\delta, \tag{4}$$

where function ϕ captures the departure of unexpectedness of item i from the preferred level of unexpectedness δ_u^* for user u and $\epsilon_{u,i}^\delta$ is the error term of this utility component for user u and item i. Then, the utility of recommending item i to user u is computed as the sum of (3) and (4):

$$U_{u,i} = U_{u,i}^q + U_{u,i}^\delta \tag{5}$$

$$U_{u,i} = q_u \times r_{u,i} - \lambda_u \times \phi(\delta_{u,i}; \delta_u^*) + \epsilon_{u,i}, \tag{6}$$

where $\epsilon_{u,i}$ is the stochastic error term. Function ϕ can also be defined in various ways. For example, using popular location models for horizontal and vertical differentiation of products in economics [Cremer and Thisse 1991; Neven 1985], the departure from the preferred level of unexpectedness can be defined as the linear distance

$$U_{u,i} = q_u \times r_{u,i} - \lambda_u \times \left|\delta_{u,i} - \delta_u^*\right|, \tag{7}$$

or the quadratic one

$$U_{u,i} = q_u \times r_{u,i} - \lambda_u \times (\delta_{u,i} - \delta_u^*)^2. \tag{8}$$

Note that the utility of a recommendation linearly increases with the rating for these distances, whereas, given the quality of the product, it increases with unexpectedness up to the threshold of the preferred level of unexpectedness δ_u^*. This threshold δ_u^* is specific for each user and context. Also, note that two recommended items of different quality and distance from the set of expected items may derive the same levels of usefulness (i.e., indifference curves).[2]

3.3. Recommendation Algorithm

Once the utility function U is defined, we can then make recommendations to user u by selecting items i having the highest values of utility $U_{u,i}$. Additionally, specific

[2] Equations (5) and (6) illustrate a simple example of a utility function for the problem of unexpectedness in recommender systems. *Any* utility function may be used and not necessarily a weighted sum of two or more distinct components. The reader might even derive examples of utility functions without the use of δ^* but may lose some of the discussed properties (e.g., global maximum). In addition, function ϕ does not have to be symmetric as in the examples provided in Equations (7) and (8).

ALGORITHM 1: Recommendation Algorithm

```
Input: Users' profiles, utility function, estimated quality of items for users, context, etc.
Output: Recommendation lists of size $N_u$

$q_{u,i}$: Quality of item $i$ for user $u$
$\underline{q}$: Lower limit on quality of recommended items
$\underline{\delta}$: Lower limit on distance of recommended items from expectations
$\bar{\delta}$: Upper limit on distance of recommended items from expectations
$N_u$: Number of items recommended to user $u$

for each user $u$ do
    Compute expectations $\mathrm{E}_u$ for user $u$;
    for each item $i$ do
        if $q_{u,i} \geq \underline{q}$ ;
        then
            Compute distance $\delta_{u,i}$ of item $i$ from expectations $\mathrm{E}_u$ for user $u$;
            if $\delta_{u,i} \in [\underline{\delta}, \bar{\delta}]$;
            then
                Estimate utility $U_{u,i}$ of item $i$ for user $u$;
            end
        end
    end
    Recommend to user $u$ top $N_u$ items having the highest utility $U_{u,i}$;
end
```

restrictions can be applied on the quality and unexpectedness of the candidate items, if appropriate in the application, to ensure that the recommended items will exhibit specific levels of unexpectedness and quality.[3]

Algorithm 1 summarizes the proposed method for generating unexpected recommendations of high quality that are hard to discover and fairly match the users' interests. In particular, we compute for each user u a set of expected recommendations E_u. Then, for each item i in our product base, if the estimated quality of the item $q_{u,i}$ is above the threshold $\underline{q}$, we compute the distance $\delta_{u,i}$ of the specific item from the set of expectations E_u for the particular user. If the distance $\delta_{u,i}$ is within the specified interval $[\underline{\delta}, \bar{\delta}]$, we compute the utility of unexpectedness $U^{\delta}_{u,i}$ of item i for user u based on $\phi(\delta_{u,i}; \delta^*_u)$. Next, we estimate the final utility $U_{u,i}$ of recommending this item to the specific user based on the different components of the specified utility function; the estimated utility corresponds to the final predicted rating $\hat{r}_{u,i}$ of the classical recommender system algorithms. Finally, we recommend to the user those items that exhibit the highest estimated utility $U_{u,i}$. Examples on how to compute the set of expected item E_u for a user are provided in Section 4.2.3.

3.4. Evaluation of Recommendations

Adomavicius and Tuzhilin [2005], Herlocker et al. [2004], and McNee et al. [2006] suggest that RSes should be evaluated not only by their accuracy, but also by other important metrics such as coverage, serendipity, unexpectedness, and usefulness. Hence, we propose specific metrics to evaluate the candidate items and the generated recommendations. To accurately and precisely measure the unexpectedness of candidate items and generated recommendation lists, we deviate from the approach proposed

[3] In the same sense, if required in a specific setting, only items not included in the set of user expectations can be considered candidates for recommendation. An alternative way to control the expected levels of unexpectedness can be based on the utility function of choice and the tuning of its coefficients.

by Murakami et al. [2008] and Ge et al. [2010] and propose new metrics to evaluate our method. In particular, Murakami et al. [2008] and Ge et al. [2010] propose an item-centric definition of unexpectedness that focuses on the difference in predictions between two algorithms (i.e., the deviation of beliefs in a recommender system from the results obtained from a primitive prediction model that shows high ratability) and thus Ge et al. [2010] calculate the unexpected set of recommendations (UNEXP) as:

$$\text{UNEXP} = \text{RS} \backslash \text{PM}, \tag{9}$$

where PM is a set of recommendations generated by a primitive prediction model and RS denotes the recommendations generated by a recommender system. When an element of RS does not belong to PM, they consider this element to be unexpected. As Ge et al. [2010] argue, unexpected recommendations may not always be useful and, thus their paper also introduces a serendipity measure as:

$$\text{SRDP} = \frac{\left|\text{UNEXP} \bigcap \text{USEFUL}\right|}{|N|}, \tag{10}$$

where N denotes the length of the recommendation list and USEFUL the set of "useful" items. For instance, the usefulness of an item can be judged by the users or approximated by the items' ratings. However, these measures do not fully capture the proposed user-centric definition of unexpectedness since a PM usually contains just the most popular items and does not actually take into account *the expectations of users*. Consequently, we revise their definition and introduce new metrics to measure unexpectedness as follows. First, we define expectedness (EXPECTED) as the mean ratio of those items that are included in both the consideration set of a user (E_u) and the generated recommendation list (RS_u):

$$\text{EXPECTED} = \sum_u \frac{\left|\text{RS}_u \bigcap \text{E}_u\right|}{|N|}. \tag{11}$$

Furthermore, we propose a metric of unexpectedness (UNEXPECTED) as the mean ratio of those items that are not included in the set of expected items for the user but are included in the generated recommendation lists:

$$\text{UNEXPECTED} = \sum_u \frac{|\text{RS}_u \backslash \text{E}_u|}{|N|}. \tag{12}$$

Correspondingly, we can also derive a new metric, following the SRDP measure of serendipity [Murakami et al. 2008], based on the proposed concept and metric of unexpectedness:

$$\text{UNEXPECTED}^{+} = \sum_u \frac{\left|(\text{RS}_u \backslash \text{E}_u) \bigcap \text{USEFUL}_u\right|}{|N|}. \tag{13}$$

For the sake of simplicity and a direct comparison with previously proposed metrics, the measures defined so far consider whether an item is expected to the user in terms of strict boolean identity. However, we can relax this restriction using the distance of an item from the set of expectations as in (1) or the unexpectedness of an item as in (2). For instance:

$$\text{UNEXPECTED} = \sum_u \frac{\Delta(\delta_{u,i}; \delta_u^*)}{|N|}. \tag{14}$$

Moreover, the metrics proposed in this section can be combined with those suggested by Murakami et al. [2008] and Ge et al. [2010] as described in Section 4.2.6. In addition, the proposed metrics can be adapted to take into consideration the rank of the item in the recommendation list by using a rank discount factor, as in Castells et al. [2011] and Vargas and Castells [2011]. Similarly, the unexpectedness of an item can be measured using different distance functions, as described in Section 4.2.4.

4. EXPERIMENTAL SETTINGS

To empirically validate the method presented in Section 3.3 and evaluate the unexpectedness of the generated recommendations, we conduct a very large number of experiments on "real-world" datasets and compare our results with those of popular baseline methods. Unfortunately, we could not compare our results with other methods for deriving unexpected recommendations for the following reasons. First, among the previously proposed methods of unexpectedness, as explained in Section 2, the corresponding authors present only performance metrics and do not provide any clear computational algorithm for computing recommendations, thus making comparison impossible. Furthermore, most of the existing methods are based on related but different principles, such as diversity and novelty. Since these concepts are, in principle, very different from our definition, they cannot be directly compared with our approach. Besides, most of the methods of novelty and serendipity require additional data, such as explicit information from users about known items. In addition, many of the methods of these related concepts are not generic and cannot be implemented in a traditional recommendation setting, but assume very specific applications and domains. Consequently, we selected a number of popular algorithms as baseline methods to compare with the proposed approach. In particular, we selected both the item-based and user-based *k*-Nearest Neighborhood approach (kNN), the Slope One (SO) algorithm [Lemire and Maclachlan 2007], a Matrix Factorization (MF) method [Koren et al. 2009], the average rating value of an item, and a baseline using the average rating value plus a regularized user and item bias [Koren 2010]. We would like to indicate that, although the selected baseline methods do not explicitly support the notion of unexpectedness, they do constitute fairly reasonable baselines because, as pointed out by Burke [2002] and Adamopoulos and Tuzhilin [2013a], CF methods also perform well in terms of other related performance measures in addition to classical accuracy measures.[4]

4.1. Datasets

The basic datasets that we used are the RecSys HetRec 2011 MovieLens (ML) dataset [Cantador et al. 2011] and the BookCrossing (BC) dataset [Ziegler et al. 2005]. The RecSys HetRec 2011 ML dataset is an extension of a dataset published by GroupLens [2011] that contains personal ratings and tags about movies and consists of 855,598 ratings from 2,113 users on 10,197 movies. This dataset is relatively dense (3.97%) compared to other frequently used datasets, but we believe that this characteristic is a virtue that will let us better evaluate our method since it allows us to better specify the set of expected movies for each user. In addition, in order to test the proposed method under various levels of sparsity [Adomavicius and Zhang 2012], we consider different proper subsets of the datasets. Additionally, we used information and further details from Wikipedia [2012] and IMDb [2011]. By joining these datasets, we were able to enhance the available information by identifying whether a movie is an episode or sequel of another movie included in our dataset. We succeeded in identifying "related" items (i.e., episodes, sequels, movies with exactly the same title) for 2,443 of our movies (23.95% of the movies, with 2.18 related movies on average and a maximum of 22). We

[4] The proposed method also outperforms in terms of unexpectedness other methods that capture the related but different concepts of novelty, serendipity, and diversity, such as the *k*-furthest neighbor CF recommender algorithm [Said et al. 2012].

used this information about related movies to identify sets of expectations, as described in Section 4.2.3. We also consider a proper subset (b) of the ML dataset consisting of 4,735 items and 2,029 users, with at least 25 ratings each, exhibiting 807,167 ratings.

The BC dataset is gathered from Bookcrossing.com [BookCrossing 2004], a social networking site founded to encourage the exchange of books. This dataset contains fully anonymized information on 278,858 members and 1,157,112 personal ratings, both implicit and explicit, referring to 271,379 distinct ISBNs. The specific dataset was selected because we can use the implicit ratings of the users to better specify their expectations, as described in Section 4.2.3. In addition, we supplemented the available data for 261,229 books with information from Amazon [2012], Google Books [2012], ISBNdb [2012], LibraryThing [2012], Wikipedia [2012], and WorldCat [2012]. Such data are often publicly available and therefore can be freely and widely used in many recommender systems [Umyarov and Tuzhilin 2011]. Since some books on BookCrossing refer to rare, non-English books or outdated titles not in print anymore, we were able to collect background information and "related" books (i.e., alternative editions, sequels, books in the same series, with same subjects and classifications, with the same tags, and books identified as related or similar by any of the aforementioned services) for 152,702 of the books, with an average of 31 related books per ISBN. Following Ziegler et al. [2005] and owing to the extreme sparsity of the BC dataset, we decided to further condense the dataset to obtain more meaningful results from CF algorithms. Hence, we discarded all books for which we were not able to find any information, along with all the ratings referring to them. Next, we also removed book titles with fewer than four ratings and community members with fewer than eight ratings each. The dimensions of the resulting dataset were considerably more modest, featuring 8,824 users, 18,607 books, and 377,749 ratings (147,403 explicit ratings). Finally, we also consider two proper subsets of this dataset: (b) 3,580 items with at least 10 ratings and 2,545 users with at least 15 ratings each, exhibiting 57,176 explicit and 95,067 implicit ratings and (c) 870 items and 1,379 users with at least 25 ratings exhibiting 22,192 explicit and 37,115 implicit ratings. Based on the collected information, we approximated the sets of expected recommendations for the users using the mechanisms described in detail in Section 4.2.3.

4.2. Experimental Setup

Using the ML dataset, we conducted 7,488 experiments. In half of the experiments, we assume that the users are homogeneous (Hom) and have exactly the same preferences. In the other half, we investigate the more realistic case (Het) where users have different preferences that depend on previous interactions with the system. Furthermore, we use two different and diverse sets of expected movies for each user and different utility functions. Also, we use different rating prediction algorithms and various measures of distance between movies and among a movie and the set of expected recommendations. Finally, we derived recommendation lists of different sizes ($k \in \{1, 3, 5, 10, 20, \ldots, 100\}$). In summary, we used two subsets, two sets of expected movies, six algorithms for rating prediction, three correlation metrics, two distance metrics, two utility functions, two different assumptions about users preferences, and 13 different lengths of recommendation lists, resulting in 7,488 experiments in total. Furthermore, using the BC dataset, we conducted our experiments on three different proper subsets described in Section 4.1. As before, we also assume different specifications for the experiments. In particular, we used three subsets, three sets of expected books, six algorithms for rating prediction, three correlation metrics, two distance metrics, two utility functions, two different assumptions about users preferences, and 13 different lengths of recommendation lists, resulting in 16,848 experiments in total. The experimental settings are described in detail in Sections 4.2.1–4.2.4.

4.2.1. Utility of Recommendation. We consider the following utility functions:

(1a) *Representative agent (homogeneous users) with linear distance* (Hom-Lin): The users are homogeneous and have similar preferences (i.e., parameters q, λ, δ^* are the same across all users) and $\phi(\delta_{u,i}; \delta^*_u)$ is linear in $\delta_{u,i}$ in Equation (6):

$$U_{u,i} = q \times r_{u,i} - \lambda \times \left|\delta_{u,i} - \delta^*\right|. \tag{15}$$

(1b) *Representative agent (homogeneous users) with quadratic distance* (Hom-Quad): The users are homogeneous but $\phi(\delta_{u,i}; \delta^*_u)$ is quadratic in $\delta_{u,i}$ in Equation (6):

$$U_{u,i} = q \times r_{u,i} - \lambda \times (\delta_{u,i} - \delta^*)^2. \tag{16}$$

(2a) *Heterogeneous users with linear distance* (Het-Lin): The users are heterogeneous, have different preferences (i.e., q_u, λ_u, δ^*_u), and $\phi(\delta_{u,i}; \delta^*_u)$ is linear in $\delta_{u,i}$ as in Equation (7):

$$U_{u,i} = q_u \times r_{u,i} - \lambda_u \times \left|\delta_{u,i} - \delta^*_u\right|. \tag{17}$$

(2b) *Heterogeneous users with quadratic distance* (Het-Quad): Users have different preferences and $\phi(\delta_{u,i}; \delta^*_u)$ is quadratic in $\delta_{u,i}$. This case corresponds to Equation (8):

$$U_{u,i} = q_u \times r_{u,i} - \lambda_u \times (\delta_{u,i} - \delta^*_u)^2. \tag{18}$$

4.2.2. Item Similarity. To generate the set of unexpected recommendations, the system computes the distance $d(i, j)$ between two items. In the conducted experiments, we use both collaborative-based and content-based item distance.[5] In addition, the computed distance matrix can be easily updated with respect to new ratings (as in Khabbaz et al. [2011]) to address potential scalability issues in large-scale systems. The complexity of the proposed algorithm can also be reduced by appropriately setting a lower limit in quality ($\underline{q}$) as illustrated in Algorithm 1. Other techniques that should also be explored in future research include user clustering, low rank approximation of the unexpectedness matrix, and partitioning the item space based on product category or subject classification.

4.2.3. Sets of Expected Recommendations. The set of expected recommendations for each user can be precisely specified and operationalized using various mechanisms that can be applied across various domains and applications. Such mechanisms include the past transactions performed by the user, knowledge discovery and data mining techniques (e.g., association rule learning and user profiling), and experts' domain knowledge. The mechanisms for specifying sets of expected recommendations for the users can also be seeded, as and when needed, with past transactions, as well as with implicit and explicit user ratings. To test the proposed method under various and diverse sets of expected recommendations of different cardinalities that have been specified using the mechanisms summarized in Table I, we consider the following settings:[6]

(1) *Expected Movies:* We use the following two examples of definitions of expected movies in our study. The first set of expected movies ($\mathrm{E}_u^{(Base)}$) for user u follows a very strict definition of expectedness, as defined in Section 3.1. The profile of user u consists of the set of movies that she or he has already rated. In particular, movie i is expected for user u if the user has already rated some movie j such that i

[5] Additional similarity measures were tested in Adamopoulos and Tuzhilin [2011] with similar results.

[6] In this experimental study, the expectations of users were specified in terms of strict boolean identity because of the characteristics of the specific datasets and for the sake of simplicity. As part of future work, we plan to relax this assumption using the proposed definition and metric of unexpectedness (Equation (14)).

Table I. Sets of Expected Recommendations for Different Experimental Settings

Dataset	Set of Expected Recommendations	Mechanism	Method
MovieLens	Base	Past Transactions	Explicit Ratings
	Base+RL	Domain Knowledge	Set of Rules
BookCrossing	Base	Past Transactions	Implicit Ratings
	Base+RI	Domain Knowledge	Related Items
	Base+AR	Data Mining	Association Rules

has the same title or is an episode or sequel of movie j, where episode or sequel is identified as explained in Section 4.1. These sets of expected recommendations have on average a cardinality of 517 and 451 for the different subsets. The second set of expected movies ($\mathrm{E}_u^{(Base+RL)}$) follows a broader definition of expectations and is generated based on some set of rules. It includes the first set plus a number of closely "related" movies ($\mathrm{E}_u^{(Base+RL)} \supseteq \mathrm{E}_u^{(Base)}$). To form the second set of expected movies, we also use content-based similarity between movies. More specifically, two movies are related if at least one of the following conditions holds: (i) they were produced by the same director, belong to the same genre, and were released within an interval of 5 years; (ii) the same set of protagonists appears in both (where a protagonist is defined as an actor with ranking $\in \{1, 2, 3\}$) and they belong to the same genre; (iii) the two movies share more than 20 common tags, are in the same language, and their correlation metric is above a certain threshold θ (Jaccard coefficient (J) > 0.50); (iv) there is a link from the Wikipedia article for movie i to the article for movie j and the two movies are sufficiently correlated ($J > 0.50$); and (v) the content-based distance metric is below a threshold θ ($d < 0.50$). The extended set of expected movies has an average size of 1,127 and 949 items per user for the two subsets, respectively.

(2) *Expected Books:* For the BC dataset, we use three different examples of expected books for our users. The first set of expectations ($\mathrm{E}_u^{(Base)}$) consists of only those items that user u rated implicitly or explicitly.[7] The second set of expected books ($\mathrm{E}_u^{(Base+RI)}$) includes the first set plus the related or similar books identified by various third-party services, as described in Section 4.1. These sets of expectations contain on average 1,257, 1,030, and 296 items for the three subsets, respectively. Finally, the third set of expected recommendations ($\mathrm{E}_u^{(Base+AS)}$) is generated using association rule learning. In detail, an item i is expected for user u if i is consequent of a rule with support of at least 5% and user u has implicitly or explicitly rated all the antecedent items. Because of the nature of this procedure, there is little variation in the set of expectations among the different users and, in general, these sets consist of the most popular items, defined in terms of number of ratings. These sets of expected recommendations have on average a cardinality of 808, 670, and 194 for the different subsets.

4.2.4. Distance from the Set of Expectations. After estimating the expectations of user u, we can then define the distance of item i from the set of expected recommendations E_u in various ways. For example, it can be determined by averaging the distances between the candidate item i and all the items included in set E_u. Additionally, we also use the centroid distance, defined as the distance of an item i from the centroid point of the set of expected recommendations E_u for user u.[8]

[7] Only explicit ratings were used with the baseline rating prediction algorithms.

[8] The experiments conducted in Adamopoulos and Tuzhilin [2011] using the Hausdorff distance ($d(i, \mathrm{E}_u) = \inf\{d(i, j) : j \in \mathrm{E}_u\}$) to estimate a lower bound of unexpectedness indicate inconsistent

4.2.5. Utility Estimation. Since users are restricted to providing ratings on a specific scale, the corresponding item ratings in our datasets are censored from below and above (also known as censoring from left and right, respectively) [Davidson and MacKinnon 2004]. Hence, to model consumer choice, estimate the parameters of interest (i.e., q_u and λ_u in Equations (15)–(18)), and make predictions within the same scale as that available to users, we borrow from the field of economics popular models of censored multiple linear regressions [McDonald and Moffitt 1980; Olsen 1978; Long 1997][9] imposing also a restriction on these models for nonnegative coefficients (i.e., $q_u, \lambda_u \geq 0$) [Greene 2012; Wooldridge 2002]. Furthermore, given the limitations of offline experiments and our datasets, we use the predicted ratings from the baseline methods as a measure of quality for the recommended items and the actual ratings of the users as a proxy for the utility of the recommendations; this, in combination with the choice of utility functions described in Section 4.2.1, allows us to study the effect of taking unexpectedness into consideration without introducing any other source of variation to our model. We also used the average distance of rated items from the set of expected recommendations to estimate the preferred level of unexpectedness δ_u^* for each user and distance metric; for the case of homogeneous users, we used the average value over all users. In addition, we did not use the unexpectedness and quality thresholds, $\underline{\delta}$, $\bar{\delta}$, and $\underline{q}$, described in Section 3.3, to limit the candidate items for recommendation. In addition, we used a hold-out validation scheme in all of our experiments with 80/20 splits of data to the training/test part to avoid overfitting. Finally, we assume an application scenario in which an item can be a candidate for recommendation to a user if and only if it has not been rated by the specific user while expected items can be recommended.

4.2.6. Metrics of Unexpectedness and Accuracy. To evaluate our approach in terms of unexpectedness, we use the metrics described in Section 3.4. Additionally, we further evaluate the recommendation lists using different (i.e., expanded) sets of expectations (compared to the expectations used for the utility estimation) based on metrics derived by combining the proposed metrics with those suggested by Murakami et al. [2008] and Ge et al. [2010]. For the primitive Prediction Model (PM) of Ge et al. [2010], in Equation (9) we used the top-N items with the highest average rating and the largest number of ratings. For instance, for the experiments conducted using the main subset of the ML dataset, the PM model consists of the top 200 items with the highest average rating and the top 800 items with the greatest number of ratings; the same ratio was used for all the experiments. In addition, we introduce an additional metric of expectedness (EXPECTED$_{\mathrm{PM}}$) as the mean ratio of the recommended items that are either included in the set of expected recommendations for a user or in the primitive prediction model and that are also included in the generated recommendation list. Correspondingly, we define an additional metric of unexpectedness (UNEXPECTED$_{\mathrm{PM}}$) as the mean ratio of the recommended items that are neither included in expectations nor in the primitive prediction model and that are included in the generated recommendations:

$$\mathrm{UNEXPECTED}_{\mathrm{PM}} = \sum_{u} \frac{|\mathrm{RS}_u \backslash (\mathrm{E}_u \cup \mathrm{PM})|}{|N|}. \tag{19}$$

Based on the ratio of Ge et al. [2010], in Equation (10), we also use the metrics UNEXPECTED$^+$ and UNEXPECTED$^+_{\mathrm{PM}}$ to evaluate serendipitous [Murakami

performance and sometimes underperformed the standard CF methods. Hence, in this work, we only conducted experiments using the average and the centroid distance.

[9] Ordered choice models and generalized linear latent and mixed models estimated by maximum likelihoods [Rabe-Hesketh et al. 2002] were also tested with similar results. Shivaswamy et al. [2007] and Khan and Zubek [2008] may also be used for utility estimation.

et al. 2008] recommendations in conjunction with the metrics of unexpectedness in Equations (12) and (19), respectively. To compute these metrics, the usefulness of an item for a user can be judged by the specific user or approximated by the item's ratings. For instance, we consider an item to be useful if its average rating is greater than the mean of the rating scale. In particular, following prior literature, in the experiments conducted using the ML and BC datasets, we consider an item to be useful if its average rating is greater than 2.5 ($\text{USEFUL} = \{i : \bar{r}_i > 2.5\}$) and 5.0, respectively. Finally, we also evaluate the generated recommendations lists based on the aggregate recommendation diversity, coverage of product base, dispersion of recommendations, and accuracy of rating and item predictions.

5. RESULTS

The aim of this study is to demonstrate that the proposed method effectively captures the concept of unexpectedness and performs well in terms of classical accuracy metrics by a comparative analysis of our method and the standard baseline algorithms in different experimental settings.

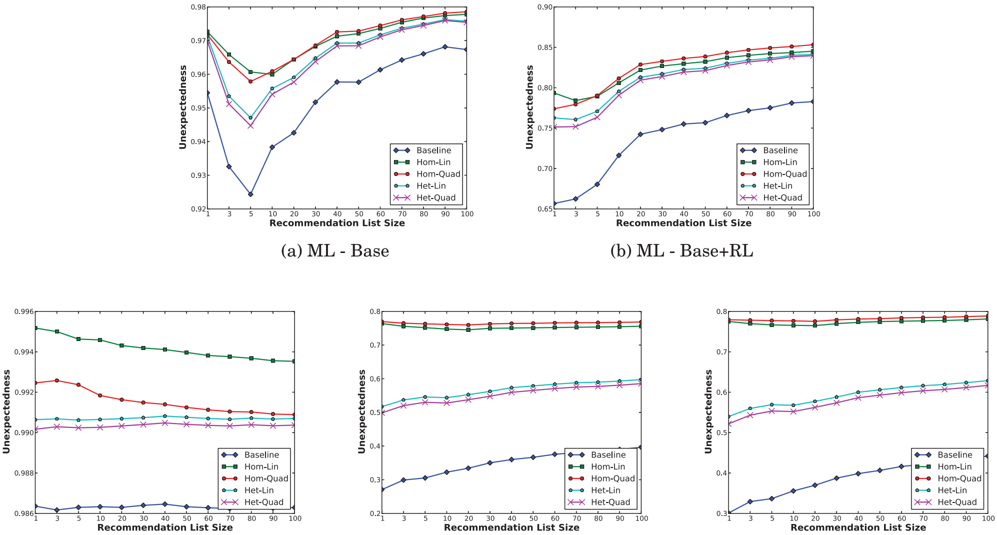
Given the number of experimental settings (five subsets based on two datasets, five sets of expected items, six algorithms for rating prediction, three correlation metrics, two distance metrics, two utility functions, two different assumptions about user preferences, and 13 different lengths of recommendation lists resulting in 24,336 experiments in total), the presentation of results constitutes a challenging problem. To give a "flavor" of the results, instead of plotting individual graphs, a more concise representation can be obtained by computing the average values of performance for the main experimental settings (see Section 4.2.1) and testing the statistical significance of the differences in performance, if any. The averages are taken over the six algorithms for rating prediction, the two correlation metrics, and the two distance metrics, except as otherwise noted. However, given the diversity of the aforementioned experimental settings, both the different baselines and the proposed approach may exhibit different performance in each setting. A reasonable way to compare the results across different experimental settings is by computing the relative performance differences:

$$\text{Diff} = (\text{Perf}_{\text{unxp}} - \text{Perf}_{\text{bsln}})/\text{Perf}_{\text{bsln}}, \tag{20}$$

taken as averages over some experimental settings, where *bsln* refers to the baseline methods and *unxp* to the proposed method for unexpectedness. A positive value of *Diff* means that the proposed method outperforms the baseline and a negative otherwise. For each metric, only the most interesting dimensions are discussed.

Using the utility estimation method described in Section 4.2.5, the average q_u is 1.005 for the experiments conducted on the ML dataset. For the experiments with the first set of expected movies, the average λ_u is 0.144 for the linear distance and 0.146 for the quadratic one. For the extended set of expected movies, the average estimated λ_u is 0.207 and 1.568, respectively. In the experiments conducted on the BC dataset, the average q_u is 1.003. For the experiments with the first set of expected books, the average λ_u is 0.710 for the linear distance and 3.473 for the quadratic one. For the second and third set of expected items, the average estimated λ_u is 0.717 and 3.1240, and 0.576 and 2.218, respectively.

In Section 5.1, we compare how the proposed method for unexpected recommendations compares with the standard baseline methods in terms of unexpectedness and serendipity of recommendation lists. Then, in Sections 5.2 and 5.3, we study the effects on rating and item prediction accuracy, respectively. Finally, in Section 5.4, we compare the proposed method with the baseline methods in terms of other popular



(a) ML - Base

(b) ML - Base+RL

(c) BC - Base

(d) BC - Base+RI

(e) BC - Base+AR

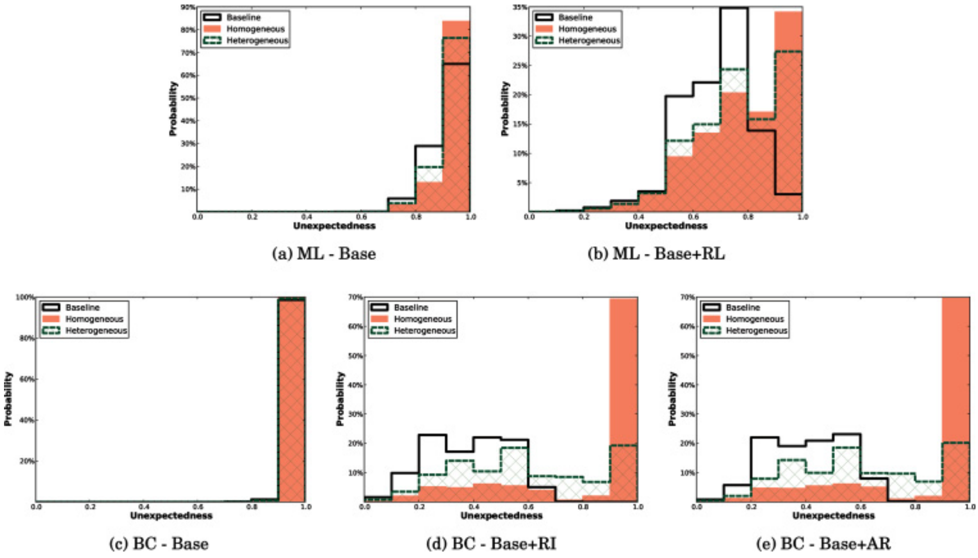
Fig. 1. Unexpectedness performance of different experimental settings for the (a), (b) MovieLens (ML) and (c), (d), (e) BookCrossing (BC) datasets.

metrics, such as catalog coverage, aggregate recommendation diversity, and dispersion of recommendations.

5.1. Comparison of Unexpectedness

In this section, we experimentally demonstrate that the proposed method effectively captures the notion of unexpectedness and thus outperforms the standard baseline methods in terms of unexpectedness. Tables VI and VIII in the online Appendix present the results obtained by applying our method to the ML and BC datasets. The values reported are computed using the proposed unexpectedness metric (Equation (12)) as the average increase in performance over six algorithms for rating prediction, two distance metrics, and three correlation metrics for recommendation lists of size $k \in \{1, 3, 5, 10, 30, 50, 100\}$. Table II summarizes these results over the different subsets. In addition, Figure 1 presents the average performance over the same dimensions for recommendation lists of size $k \in \{1, 3, 5, 10, 20, \ldots, 100\}$. Similar results were also obtained using the additional metrics described in Section 4.2.6. In addition, similar patterns were also observed specifying the user expectations using different mechanisms for the training and test data.

Tables II, VI, and VIII, as well as Figure 1, demonstrate that the proposed method outperforms the standard baselines. As we can observe, the increase in performance is larger for recommendation lists of smaller size k. This, in combination with the observation that unexpectedness was significantly enhanced also for large values of k, illustrates that the proposed method both introduces new items in the recommendation lists and also effectively re-ranks existing items by promoting unexpected ones. Figure 1 also shows that unexpectedness was enhanced both in cases where the definition of unexpectedness was strict, as described in Section 4.2.3—and thus the baseline recommendation system methods resulted in high unexpectedness (i.e., Base)—and in cases where the measured unexpectedness of the baselines was low (i.e., Base+RL, Base+RI, and Base+AR). Similarly, as Figures 8 and 9 show, the performance was increased both for the baseline methods that resulted in high unexpectedness (e.g., SO



(a) ML - Base

(b) ML - Base+RL

(c) BC - Base

(d) BC - Base+RI

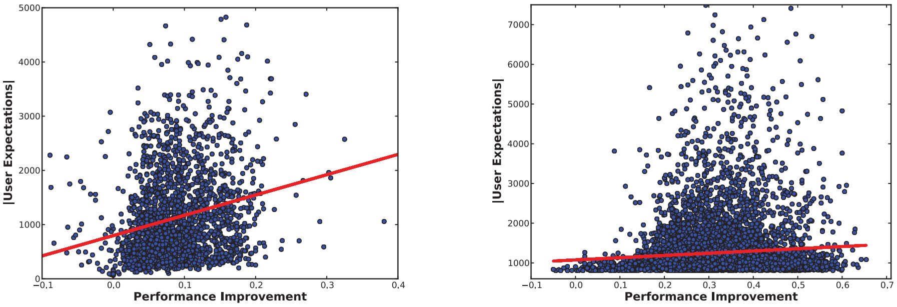
(e) BC - Base+AR

Fig. 2. Distribution of Unexpectedness for recommendation lists of size $k = 5$ and different experimental settings for the MovieLens (ML) and BookCrossing (BC) datasets.

algorithm) in the conducted experiments and for the methods where unexpectedness was low (e.g., MF method, item-based kNN recommendation algorithm).[10] Additionally, the experiments conducted using the more accurate sets of expectations based on the information collected from various third-party websites (Base+RI) outperformed those automatically derived by association rules (Base+AS). In addition, Tables VI and VIII indicate that the increase in performance is larger also in those experiments where the sparsity of the subset of data (see Section 4.1) is higher, which is the most realistic scenario in practice. In particular, for the ML dataset, the average unexpectedness of the recommendation lists was increased by 1.62% and 10.83% (17.32% for $k = 1$) for the (Base) and (Base+RL) sets of expected movies, respectively. For the BC dataset, for the (Base) set of expectations, the average unexpectedness was increased by 0.55%. For the (Base+RI) and (Base+AR) sets of expected books, the average improvement was 135.41% (188.61% for $k = 1$) and 78.16% (117.28% for $k = 1$). Unexpectedness was increased in 85.43% and 89.14% of the experiments for the ML and BC datasets, respectively. Finally, the unexpectedness of the generated recommendation lists can be further enhanced, as described in Section 3.3, using appropriate thresholds on the unexpectedness of individual items.

A particularly noteworthy observation, as demonstrated through the distribution of unexpectedness across all the generated recommendation lists for the ML and BC datasets in Figure 2, is that the higher the cardinality and the better approximated the sets of users' expectations, the greater the improvements against the baseline methods.[10] In principle, if no expectations are specified, the recommendation results will be the same as the baseline method. The same pattern can also be observed in Figure 3, showing the cardinality of the set of user expectations along the vertical axis, the increase in unexpectedness performance along the horizontal axis, and a linear

[10] Figures 8 and 9 in the online Appendix present the distribution of unexpectedness across all the users for the different rating estimation algorithms using the ML and BC datasets with the respective sets of user expectations (Base+RL) and (Base+RI), and recommendation lists of size $k = 5$.



(a) ML - Base+RL

(b) BC - Base+RI

Fig. 3. Increase in Unexpectedness for recommendation lists of size $k = 5$ for the MovieLens (ML) and BookCrossing (BC) datasets using different sets of expectations.

line fitting the data for recommendation lists of size $k = 5$.[11] This informal notion of "monotonicity" of expectations is useful in order to achieve the desired levels of unexpectedness. We believe that this pattern is a general property of the proposed method because of the explicit use of users' expectations and the departure function, and we plan to explore this topic as part of our future research.

To determine statistical significance, we tested the null hypothesis that the performance of each of the five lines of the graphs in Figure 1 is the same, using the Friedman test (nonparametric repeated measure ANOVA) [Berry and Linoff 1997], and we reject the null hypothesis with $p < 0.0001$. Performing post hoc analysis on Friedman's test results for the ML dataset, the difference between the *Baseline* and each one of the experimental settings, apart from the difference between the *Baseline* and *Heterogeneous Quadratic*, are statistically significant. In addition, the differences between *Homogeneous Quadratic* and *Heterogeneous Linear*, *Homogeneous Linear* and *Heterogeneous Quadratic*, and *Homogeneous Quadratic* and *Heterogeneous Quadratic* are statistically significant as well. For the BC dataset, the difference between the *Baseline* and each one of the experimental settings is also statistically significant, with $p < 0.0001$. Moreover, the differences among *Homogeneous Linear*, *Homogeneous Quadratic*, *Heterogeneous Linear*, and *Heterogeneous Quadratic*, apart from the difference between *Homogeneous Linear* and *Homogeneous Quadratic*, are also statistically significant.

5.1.1. Qualitative Comparison of Unexpectedness. The proposed approach avoids obvious recommendations such as recommending to a user the movies "The Lord of the Rings: The Return of the King," "The Bourne Identity," and "The Dark Knight" after the user had already highly rated all the sequels or prequels of these movies. In addition, the proposed method provides recommendations from a wider range of items and does not focus mostly on bestsellers, as described in Section 5.4. In addition, even though the proposed method generates truly unexpected recommendations, these recommendations are not irrelevant and they still provide a fair match to user's interests. Finally, to further evaluate the proposed approach, we present some examples of recommendations; additional examples for each set of expectations are presented in Section A.1 of the online Appendix.

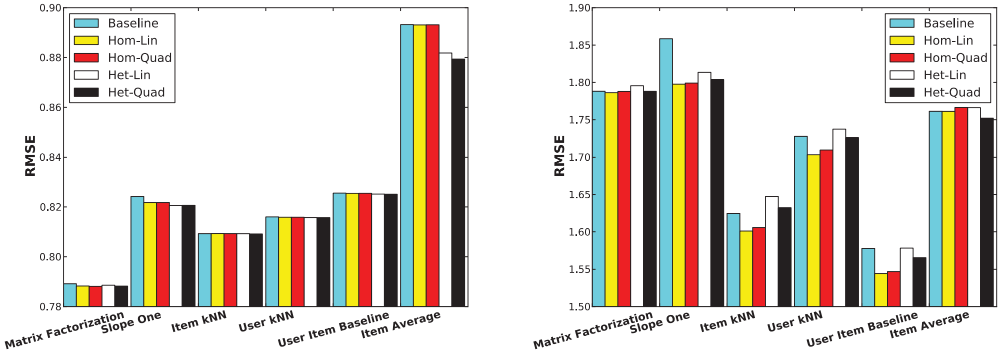
Using the ML dataset and the (Base) sets of expected recommendations, the baseline methods recommend to a user who highly rates very popular Action, Adventure, and Drama films the movies "The Lord of the Rings: The Two Towers," "The Dark Knight," and "The Lord of the Rings: The Return of the King" (user id = 36803 with MF).

[11] We also tried higher order polynomials, but they do not offer significantly better fitting of the data.

However, this user has already highly rated prequels or sequels of these movies (i.e., "The Lord of the Rings: The Fellowship of the Ring" and "Batman Begins"); thus, the aforementioned popular recommendations are expected for this specific user. On the other hand, for the same user, the proposed method generated the following recommendations: "The Pianist," "La vita è bella," and "Rear Window." These movies are of high quality, unexpected, and not irrelevant, since they fairly match the user's interests. In particular, based on the definitions and mechanisms used to specify user expectations as described in Section 4.2.3, all these interesting movies are unexpected for the user since they significantly depart from her or his expectations. Additionally, they are of great quality in terms of the average rating, even though less popular in terms of the number of ratings. In addition, these Biography, Drama, Romance, and Mystery movies are not irrelevant to the user, and they fairly match the user's profile since they involve elements in their plot (such as war) that can also be found in other films that she or he has already highly rated (such as "Erin Brockovich," "October Sky," and "Three Kings"). Finally, interestingly enough, some of these high-quality, interesting, and unexpected recommendations are also based on movies filmed by the same director who adapted a film the user rated highly (i.e., "Pinocchio" and "La vita è bella").

Using the BC dataset and the (Base+RI) set of expectations described in Section 4.2.3, the baseline methods recommend to a user who has already rated a very large number of items the following expected books: "I Know This Much Is True," "Outlander," and "The Catcher in the Rye" (user id = 153662 with MF). In particular, the book "I Know This Much Is True" is highly expected because the specific user has already rated and is familiar with the books "A Tangled Web," "A Virtuous Woman," "Thursday's Child," and "Drowning Ruth." Similarly, the book "Outlander" is expected because of the books "Dragonfly in Amber," "Enslaved," "When Lightning Strikes," "Touch of Enchantment," and "Thorn in My Heart." Finally, the recommendation about the item "The Catcher in the Rye" is expected since the user has highly rated the books "Forever: A Novel of Good and Evil, Love and Hope," "Fahrenheit 451," and "Dream Country." In summary, all of the aforementioned recommendations are expected for the user because the recommended items are very similar to other books that the user has already highly rated from the same authors that were published around the same time (e.g., "I Know This Much Is True" and "A Virtuous Woman," or "Outlander" and "Dragonfly in Amber," etc.); frequently bought together on popular websites such as Amazon.com [Amazon 2012] and LibraryThing [LibraryThing 2012] (e.g., "I Know This Much Is True" and "Drowning Ruth," etc.); with similar library subjects, plots, and classifications (e.g., "The Catcher in the Rye" and "Dream Country," etc.); with similar tags (e.g., "The Catcher in the Rye" and "Forever: A Novel of Good and Evil, Love and Hope"), and the like. Despite that, the proposed algorithm recommends to the user the following books that significantly depart from her or his expectations: "Doing Good," "The Reader," and "Tuesdays with Morrie: An Old Man, a Young Man, and Life's Greatest Lesson." These high-quality and interesting recommendations, even though unexpected to the user, are not irrelevant since they provide a fair match to the user's interests as she or he has already highly rated books that deal with relevant issues such as family, romance, life, and memoirs.

5.1.2. Comparison of Serendipity. Pertaining to the notion of serendipity as defined in Ge et al. [2010], Tables VII and IX in the online Appendix present the results obtained by applying our method to the ML and BC datasets. The values reported are computed using the adapted metric (Equation (13)) as the average increase in performance over six algorithms for rating prediction, two distance metrics, and three correlation metrics for recommendation lists of size $k \in \{1, 3, 5, 10, 30, 50, 100\}$. Figure 10 presents the average performance recommendation lists of size $k \in \{1, 3, 5, 10, 20, \ldots, 100\}$. Similar



(a) ML - RMSE

(b) BC - RMSE

Fig. 4. RMSE performance for the (a) MovieLens and (b) BookCrossing datasets.

results were also obtained using the supplementary metrics described in Section 4.2.6, including the metrics suggested by Murakami et al. [2008] and Ge et al. [2010] and the additionally proposed metrics.

In summary, we demonstrated in this sections that *the proposed method for unexpected recommendations effectively captures the notion of unexpectedness by providing the users with interesting and unexpected recommendations of high quality that fairly match their interests* and thus outperforms the standard baseline methods in terms of the proposed unexpectedness metrics.

5.2. Comparison of Rating Prediction

In this section, we examine how the proposed method for unexpected recommendations compares with the standard baseline methods in terms of classical rating prediction accuracy-based metrics, such as RMSE and MAE. In typical offline experiments as those presented here, the data are not collected using the recommender system or method under evaluation. In particular, the observations in our test sets were not based on unexpected recommendations generated from the proposed method.[12] Also, the user ratings had been submitted over a long period of time and represented the tastes of users and their expectations of the recommender system at that specific point in time when they rated each item. Therefore, to effectively evaluate the rating and item prediction accuracy of our method, when we compute the unexpectedness of item i for user u (see Section 3.3), we treat item i as not being included in the set of expectations E_u for user u —whether it is included or not— and we compute the distance of item i from the rest of the items in the set of expectations E_u^{-i}, where $\mathrm{E}_u^{-i} := \mathrm{E}_u \backslash \{i\}$, to generate the corresponding prediction $\hat{r}_{u,i}$ (i.e., the estimated utility of recommending the candidate item i to the target user u).

Tables X–XIII in the online Appendix present the results obtained by applying our method to the ML and BC datasets using the different sets of expectations and baseline predictive methods. The values reported are computed as the difference in average performance over the different utility functions, two distance metrics, and three correlation metrics. Table III summarizes these results over the different subsets for the RMSE. In Figure 4, the bars labeled *Baseline* represent the performance of the standard baseline

[12] For instance, the assumption that unused items would have not been used even if they had been recommended is erroneous when you evaluate unexpected recommendations (i.e., a user may not have used an item because she or he was unaware of its existence, but, after the recommendation exposed that item, the user can decide to select it [Shani and Gunawardana 2011]).

methods. The bars labeled *Homogeneous Linear*, *Homogeneous Quadratic*, *Heterogeneous Linear*, and *Heterogeneous Quadratic* present the average performance over the different subsets and sets of expectations, two distance metrics, and three correlation metrics for the different experimental settings described in Section 4.2.1. All the bars have been grouped by baseline algorithm (x-axis).

In the aforementioned tables and figures, we observe that the proposed method performs at least as well as the standard baseline methods in most of the experimental settings. In particular, for the ML dataset, the RMSE was on average reduced by 0.07% and 0.34% for the cases of the homogeneous and heterogeneous users. For the BC dataset, the RMSE was improved by 1.30% and 0.31%, respectively. The overall minimum average RMSE achieved was 0.7848 for the ML and 1.5018 for the BC dataset.

Using the Friedman test, we tested the null hypothesis that the performance of each of the five lines of the graphs in Figure 4 is the same; we reject the null hypothesis with $p < 0.001$. Performing post hoc analysis on Friedman's test results, for the ML dataset only, the difference between the *Heterogeneous Quadratic* and *Baseline* is statistically significant for the RMSE accuracy metric. For the BC dataset, the differences between the *Homogeneous Linear* and *Baseline*, and *Homogeneous Quadratic* and *Baseline* are statistically significant, as well.

In summary, we demonstrated in this section that the proposed method performs at least as well as, and in some cases even better than, the standard baseline methods in terms of the classical rating prediction accuracy-based metrics.

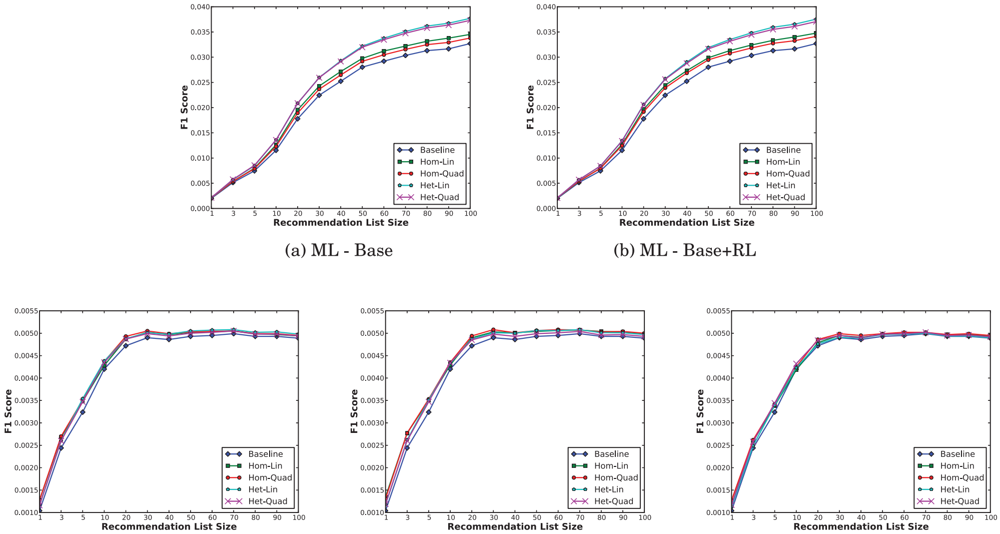
5.3. Comparison of Item Prediction

The goal in this section is to compare our method with the standard baseline methods in terms of traditional metrics for item prediction, such as precision, recall, and F_1 score. Table IV in the Appendix presents the results obtained by applying our method to the ML and BC datasets. The values reported are computed as the difference in average performance over the different subsets, six algorithms for rating prediction, two distance metrics, and three correlation metrics using the F_1 score for recommendation lists of size $k \in \{1, 3, 5, 10, 30, 50, 100\}$. Respectively, Figure 5 illustrates the average performance over the same dimensions for lists of size $k \in \{1, 3, 5, 10, 20, \ldots, 100\}$.

In particular, for the ML dataset and the case of the homogeneous users, F_1 score was improved by 6.14%, on average. In the case of heterogeneous customers, performance was increased by 13.90%. For the BC dataset, in the case of homogeneous users, the F_1 score was on average enhanced by 4.85% and for heterogeneous users by 3.16%.[13] Table IV shows that performance was increased both in cases where the definition of unexpectedness was strict (i.e., Base) and in cases where the definition was broader (i.e., Base+RL, Base+RI, and Base+AR). Additionally, the experiments conducted using the more accurate sets of expectations based on the information collected from various third-party websites (Base+RI) outperformed those using the expected sets automatically derived by association rules (Base+AS).

To determine statistical significance, we tested the null hypothesis that the performance of each of the five lines of the graphs in Figure 5 is the same using the Friedman test. Based on the results, we reject the null hypothesis with $p < 0.0001$. Performing post hoc analysis on Friedman's test results for the ML dataset, the differences between the *Baseline* and each one of the experimental settings are statistically significant for the F_1 score. For the BC dataset, the differences between the *Baseline* and each one of

[13] In Tables XIV–XVII of the online Appendix, detailed results for precision and recall are presented as well.



(a) ML - Base

(b) ML - Base+RL

(c) BC - Base

(d) BC - Base+RI

(e) BC - Base+AR

Fig. 5. F_1 performance of different experimental settings for the (a), (b) MovieLens (ML) and (c), (d), (d) BookCrossing (BC) datasets.

the experimental settings are also statistically significant.[14] Even though the lines are very close to each other and the differences in performance in absolute values are not large (e.g., Figure 5(e)), the results are statistically significant since the performance of the proposed method is ranked consistently higher than the baselines (lines do not cross).

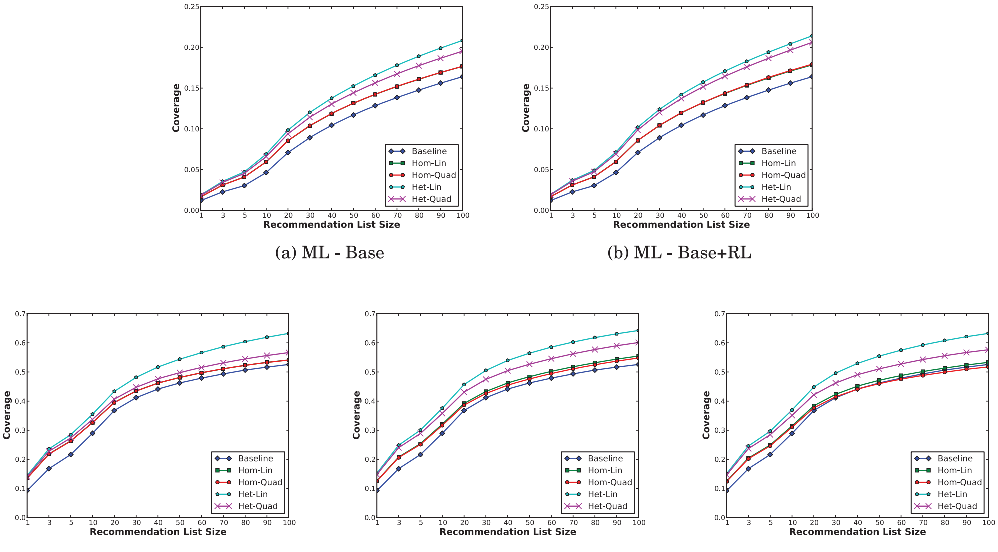
In conclusion, we demonstrated in this section that the proposed method for unexpected recommendations performs at least as well as, and in some cases even better than, the standard baseline methods in terms of the classical item prediction metrics.

5.4. Comparison of Catalog Coverage, Aggregate Recommendation Diversity, and Dispersion of Recommendations

In this section, we investigate the effect of the proposed method for unexpected recommendations on coverage, aggregate diversity, and dispersion, three important metrics for RSes [Ge et al. 2010; Adomavicius and Kwon 2012; Shani and Gunawardana 2011].[15] The results obtained using the *catalog coverage* metric [Herlocker et al. 2004; Ge et al. 2010] (i.e., the percentage of items in the catalog that are ever recommended to users: $|\bigcup_{\mathrm{u}\in\mathrm{U}} \mathrm{RS}_u|/|I|$) are very similar to those using the *diversity-in-top-N* metric for aggregate diversity [Adomavicius and Kwon 2011, 2012]; henceforth, only results on coverage are presented. Tables XVIII and XIX in the online Appendix present the results obtained by applying our method to the ML and BC datasets. The values reported are computed as the average catalog coverage over six algorithms for rating prediction, two distance metrics, and three correlation metrics for recommendation lists of size $k \in \{1, 3, 5, 10, 30, 50, 100\}$. Table V in the Appendix summarizes these results over the

[14] In the experiments conducted using the ML dataset, the difference between *Homogeneous Quadratic* and *Baseline* is statically significant with $p < 0.01$.

[15] High unexpectedness of recommendation lists does not imply high coverage and diversity. For example, if the system recommends to all users the same k best unexpected items from the product base, the recommendation list for each user is unexpected, but only k distinct items are recommended to all users.



(a) ML - Base

(b) ML - Base+RL

(c) BC - Base

(d) BC - Base+RI

(e) BC - Base+AR

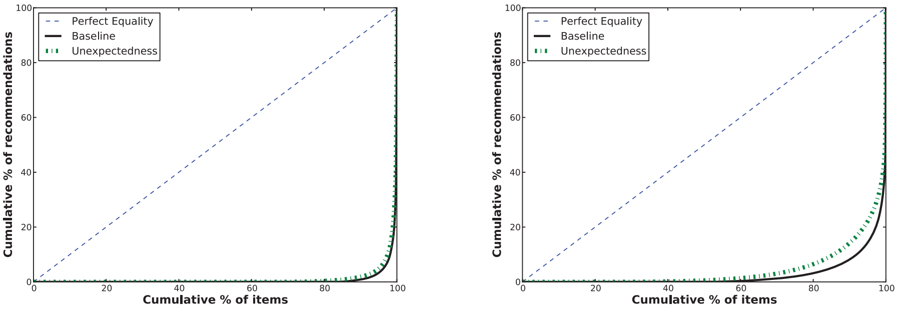
Fig. 6. Coverage performance of different experimental settings for the (a), (b) MovieLens (ML) and (c), (d), (e) BookCrossing (BC) datasets.

different subsets. Figure 6 presents the average performance over the same dimensions for recommendation lists of size $k \in \{1, 3, 5, 10, 20, \ldots, 100\}$.

As Table V, XVIII, and XIX and Figure 6 demonstrate, the proposed method outperforms the standard baselines in most of the experimental settings. As we can see, the experiments conducted under the assumption of heterogeneous users exhibit higher catalog coverage than those using a representative agent. This is an interesting result that can be useful in practice, especially in settings with potential adverse effects of over-recommending an item or very large catalogs. For instance, it would be profitable for Netflix [2012], if the recommender system could encourage users to rent "long-tail" movies, given that such movies are less costly to license and acquire from distributors than the new-release or highly popular movies of big studios [Goldstein and Goldstein 2006]. Also, we can observe that the smaller the size of the recommendation list, the greater the increase in performance. In particular, as we see in Table V, for the ML dataset, the average coverage was increased by 19.48% (39.10% for $k = 1$) and 37.40% (58.39% for $k = 1$) for the cases of the homogeneous and heterogeneous users, respectively. For the BC dataset, in the case of homogeneous customers, coverage was improved by 9.26% (39.00% for $k = 1$) and for heterogeneous customers by 23.17% (59.62% for $k = 1$), on average. In addition, Tables XVIII and XIX illustrate that the increase in performance is larger also in the experiments where the sparsity of the subset of data is higher. In general, coverage was increased in 95.68% (max = 55.74%) and 91.57% (max = 100%) of the experiments for the ML and BC datasets, respectively.

In terms of statistical significance, with the Friedman test, we have rejected the null hypothesis ($p < 0.0001$) that the performance of each of the five lines of the graphs in Figure 6 is the same. Performing post hoc analysis on Friedman's test results, for both the datasets, the difference between the *Baseline* and each of the remaining experimental settings is statistically significant ($p < 0.001$).

The derived recommendation lists can also be evaluated for the inequality across items and the dispersion of recommendations, using the Gini coefficient [Gini 1909], the Hoover (Robin Hood) index [Hoover 1985], or the Lorenz curve [Lorenz 1905]. In



(a) ML - Base+RL

(b) BC - Base+RI

Fig. 7. Lorenz curves for recommendation lists of size $k = 5$ for the (a) MovieLens (ML) and (b) BookCrossing (BC) datasets.

particular, Figure 7 uses the Lorenz curve to graphically represent the cumulative distribution function of the empirical probability distribution of recommendations; it is a graph showing for the bottom x% of items what percentage y% of the total recommendations they have. As we can conclude from Figure 7, in the recommendation lists generated from the proposed method, the number of times an item is recommended is more equally distributed compared to the baseline methods. Such systems provide recommendations from a wider range of items and do not focus mostly on bestsellers, which users are often capable of discovering by themselves. Hence, they are beneficial for both users and some organizations [Brynjolfsson et al. 2003, 2011; Goldstein and Goldstein 2006]. Finally, the difference in increase in performance between Figures 7(a) and 7(b) —0.98% and 7.17%, respectively, in terms of the Hoover index— could be attributed to both idiosyncrasies of the two datasets and to the differences in definitions and cardinalities of the sets of expected recommendations discussed in Section 4.2.3.

In summary, we demonstrated in this section that the proposed method for unexpected recommendations outperforms the standard baseline methods in terms of the classical catalog coverage measure, aggregate recommendation diversity, and dispersion of recommendations.

6. DISCUSSION AND CONCLUSION

In this article, we proposed a method to improve user satisfaction by generating unexpected recommendations based on the utility theory of economics. In particular, we proposed and studied a new concept of unexpected recommendations as recommending to a user those items that depart from what the specific user expects from the recommender system—the consideration set of the user. We defined and formalized the concept of unexpectedness and discussed how it differs from the related notions of novelty, serendipity, and diversity. In addition, we suggested several mechanisms for specifying the users' expectations and proposed specific performance metrics to measure the unexpectedness of recommendation lists. After formally defining and theoretically formulating this concept, we operationalized the notion of *unexpectedness* and presented a method for providing unexpected recommendations of high quality that are hard to discover but fairly match users' interests.

Moreover, we compared the generated unexpected recommendations with popular baseline methods using the proposed performance metrics of unexpectedness. Our experimental results demonstrate that the proposed method improves performance in terms of unexpectedness while maintaining the same or higher levels of accuracy of recommendations. In addition, we showed that the proposed method for unexpected recommendations also improves performance based on other important metrics, such as catalog coverage, aggregate diversity, and dispersion of recommendations. More specifically, using different "real-world" datasets, various examples of sets of expected recommendations, and different utility functions and distance metrics, we were able to test the proposed method under a large number of experimental settings including various levels of sparsity, different mechanisms for specifying users' expectations, and different cardinalities of these sets of expectations. As discussed in Section 5, all the examined variations of the proposed method, including homogeneous and heterogeneous users with different departure functions, both introduce new unexpected items in the recommendation lists and effectively promote the existing unexpected ones and thus significantly outperformed in terms of unexpectedness the standard baseline algorithms, including item-based and user-based *k*NN, SO [Lemire and Maclachlan 2007], and MF [Koren et al. 2009]. This demonstrates that the proposed method indeed effectively captures the concept of unexpectedness since, in principle, it should do better than unexpectedness-agnostic methods such as the classical CF approach. Furthermore, the proposed unexpected recommendation method performed at least as well as, and in some cases even better than, the baseline algorithms in terms of classical accuracy-based measures, such as RMSE and F_1 score.

One of the main premises of the proposed method is that users' expectations should be explicitly considered in order to provide users with unexpected recommendations of high quality that are hard to discover but fairly match their interests. If no expectations are specified, the recommendation results will not differ from those of the standard rating prediction algorithms in recommender systems. Hence, the greatest improvements both in terms of unexpectedness and accuracy vis-à-vis all other approaches were observed in those experiments using the sets of expectations exhibiting larger cardinality (Base+RL, Base+RI, and Base+AS). These sets of expected recommendations allowed us to better approximate the expectations of each user through a non-restricting but more realistic and natural definition of "expected" items using the particular characteristics of the selected datasets (see Section 4.1). Additionally, the experiments conducted using the more accurate sets of expectations based on the information collected from various third-party websites (Base+RI) outperformed those using the expected sets automatically derived by association rules (Base+AS). Also, the fact that the proposed method delivers unexpected recommendations of high quality is depicted in the small differences between the proposed metric of unexpectedness (Equation (12)) and the adapted metric of serendipity (Equation (13)) illustrated in Tables VI–IX.

Moreover, the standard example of a utility function that was provided in Section 3.2 illustrates that the proposed method can be easily used in existing recommender systems as a new component that enhances unexpectedness of recommendations without the need to modify the current rating prediction procedures. Furthermore, since the proposed method is not specific to the examples of utility functions and sets of expected recommendations that were provided in this work, we suggest adapting the proposed method to particular recommendation applications by experimenting with different utility functions, estimation procedures, and sets of expectations, thus exploiting the domain knowledge. Similarly, the proposed approach can be easily extended to take advantage of the multidimensionality of users' profiles and tastes by employing multiple sets of expectations for each user.

The proposed approach also has important managerial implications. By avoiding obvious and expected recommendations while maintaining high predictive accuracy levels [Adamopoulos and Tuzhilin 2013b], we can alleviate the common problems of overspecialization and concentration bias, which often characterize CF algorithms [Adamopoulos and Tuzhilin 2014], and this will lead to further increases in both user satisfaction and engagement [Baumol and Ide 1956]. In addition, introducing unexpectedness in RSes, we can improve the welfare of consumers by allowing them to locate and buy better products that they would not have purchased otherwise and vastly reduce customers' search cost by recommending items that would be quite unlikely or time-consuming to discover. As a result, the inefficiencies caused by buyer search costs are reduced while simultaneously increasing the ability of markets to optimally allocate productive resources [Bakos 1997]. In addition, the proposed approach also exhibits a positive economic effect for businesses based on increased sales and willingness-to-pay [Brynjolfsson et al. 2003], additional revenues from market niches that usually exhibit lower marginal costs and higher profit margins [Fleder and Hosanagar 2009], and enhanced customer loyalty leading to lasting and valuable relationships [Gorgoglione et al. 2011].

As a part of our future work, we would like to conduct live experiments with real users in order to evaluate the unexpected ness of recommendations and analyze both qualitative and quantitative aspects in a traditional online retail setting, as well as in a platform for massive open online courses [Adamopoulos 2013b]. Moreover, we will further explore the notion of "monotonicity" introduced in Section 5.1 with the goal of formally and empirically demonstrating this effect. Furthermore, we assumed in all the experiments reported in this article that a recommendation can be either expected or unexpected. We plan to relax this assumption in our future experiments using the proposed definition and metrics of unexpectedness. Finally, we would also like to introduce and study additional metrics of unexpectedness and further investigate how the different existing recommender system algorithms perform in terms of unexpectedness vis-à-vis other popular properties of recent systems.

APPENDIX

A. UNEXPECTEDNESS

Table II. Increase in Unexpectedness Performance for the MovieLens and BookCrossing Datasets

Data set	User Expectations	Experimental Setting	Recommendation List Size						
			1	*3*	*5*	*10*	*30*	*50*	*100*
MovieLens	Base	Homogeneous Linear	1.90%	3.57%	3.93%	2.30%	1.74%	1.51%	1.08%
		Homogeneous Quadratic	1.81%	3.33%	3.63%	2.40%	1.77%	1.58%	1.16%
		Heterogeneous Linear	1.77%	2.24%	2.46%	1.86%	1.37%	1.21%	0.87%
		Heterogeneous Quadratic	1.61%	1.99%	2.21%	1.68%	1.27%	1.13%	0.84%
	Base+RL	Homogeneous Linear	20.84%	18.37%	16.01%	12.53%	10.51%	9.98%	7.97%
		Homogeneous Quadratic	17.86%	17.67%	16.14%	13.31%	11.28%	10.82%	8.99%
		Heterogeneous Linear	16.14%	14.82%	13.28%	11.06%	9.22%	8.90%	7.46%
		Heterogeneous Quadratic	14.43%	13.50%	12.20%	10.39%	8.76%	8.51%	7.26%
BookCrossing	Base	Homogeneous Linear	0.89%	0.90%	0.84%	0.84%	0.79%	0.77%	0.73%
		Homogeneous Quadratic	0.62%	0.65%	0.62%	0.56%	0.52%	0.50%	0.47%
		Heterogeneous Linear	0.43%	0.46%	0.44%	0.44%	0.44%	0.45%	0.45%
		Heterogeneous Quadratic	0.39%	0.42%	0.40%	0.40%	0.41%	0.41%	0.41%
	Base+RI	Homogeneous Linear	182.12%	152.70%	146.17%	131.80%	114.17%	104.80%	90.69%
		Homogeneous Quadratic	184.29%	155.78%	149.89%	136.12%	117.89%	108.54%	93.88%
		Heterogeneous Linear	91.03%	79.54%	78.75%	68.62%	60.64%	57.82%	50.74%
		Heterogeneous Quadratic	84.19%	73.90%	73.57%	63.73%	56.53%	54.18%	47.69%
	Base+AR	Homogeneous Linear	157.56%	133.80%	127.74%	115.27%	98.71%	90.49%	76.75%
		Homogeneous Quadratic	158.95%	136.38%	130.90%	118.38%	101.16%	92.43%	78.44%
		Heterogeneous Linear	79.30%	70.04%	69.09%	59.62%	51.84%	49.09%	42.22%
		Heterogeneous Quadratic	73.31%	64.99%	64.44%	55.24%	48.17%	45.86%	39.57%

Note: Recommendation lists of size $k \in \{20, 40, 60, 70, 80, 90\}$ were not included because of space limitations.

B. RATING PREDICTION

Table III. Average RMSE Performance for the MovieLens and BookCrossing Datasets

Data set	Rating Prediction Algorithm	User Expectations	Baseline	Homogeneous		Heterogeneous	
				Linear	Quadratic	Linear	Quadratic
MovieLens	MatrixFactorization	Base	0.7892	0.11%	0.13%	0.07%	0.12%
		Base+RL	0.7892	0.12%	0.13%	0.07%	0.12%
	SlopeOne	Base	0.8242	0.29%	0.29%	0.43%	0.43%
		Base+RL	0.8242	0.29%	0.29%	0.43%	0.42%
	ItemKNN	Base	0.8093	−0.01%	−0.01%	0.00%	0.01%
		Base+RL	0.8093	−0.01%	−0.01%	0.01%	0.02%
	UserKNN	Base	0.8160	0.01%	0.01%	0.03%	0.04%
		Base+RL	0.8160	0.01%	0.01%	0.03%	0.04%
	UserItemBaseline	Base	0.8256	0.01%	0.00%	0.04%	0.05%
		Base+RL	0.8256	0.01%	0.01%	0.06%	0.05%
	ItemAverage	Base	0.8932	0.01%	0.00%	1.26%	1.52%
		Base+RL	0.8932	0.02%	0.01%	1.29%	1.57%
BookCrossing	MatrixFactorization	Base	1.7882	0.28%	0.35%	−0.35%	0.02%
		Base+RI	1.7882	0.05%	−0.14%	−0.42%	0.01%
		Base+AS	1.7882	0.01%	−0.14%	−0.46%	−0.01%
	SlopeOne	Base	1.8585	3.43%	3.52%	2.58%	3.12%
		Base+RI	1.8585	3.15%	3.01%	2.32%	2.79%
		Base+AS	1.8585	3.21%	3.04%	2.37%	2.91%
	ItemKNN	Base	1.6248	1.46%	1.45%	−1.21%	−0.23%
		Base+RI	1.6248	1.43%	1.02%	−1.44%	−0.59%
		Base+AS	1.6248	1.48%	1.02%	−1.52%	−0.54%
	UserKNN	Base	1.7280	1.41%	1.19%	−0.41%	0.25%
		Base+RI	1.7280	1.44%	0.99%	−0.66%	−0.02%
		Base+AS	1.7280	1.46%	1.01%	−0.60%	0.10%
	UserItemBaseline	Base	1.5779	2.48%	2.34%	0.21%	0.99%
		Base+RI	1.5779	1.93%	1.77%	−0.14%	0.68%
		Base+AS	1.5779	1.98%	1.78%	−0.14%	0.71%
	ItemAverage	Base	1.7615	0.07%	−0.10%	−0.17%	0.50%
		Base+RI	1.7615	−0.04%	−0.32%	−0.28%	0.56%
		Base+AS	1.7615	0.01%	−0.41%	−0.35%	0.50%

C. ITEM PREDICTION

Table IV. Increase in F_1 Performance for the MovieLens and BookCrossing Datasets

Data set	User Expectations	Experimental Setting	Recommendation List Size						
			1	*3*	*5*	*10*	*30*	*50*	*100*
MovieLens	Base	Homogeneous Linear	5.00%	4.29%	7.10%	9.54%	8.15%	6.17%	5.57%
		Homogeneous Quadratic	4.00%	4.87%	5.63%	6.68%	5.35%	4.10%	3.36%
		Heterogeneous Linear	5.00%	10.92%	13.67%	17.78%	15.63%	14.81%	15.29%
		Heterogeneous Quadratic	7.50%	12.09%	14.61%	17.78%	15.50%	14.09%	14.07%
	Base+RL	Homogeneous Linear	3.00%	4.48%	7.37%	10.15%	8.78%	6.64%	6.33%
		Homogeneous Quadratic	4.50%	5.46%	6.70%	7.98%	6.55%	5.14%	4.37%
		Heterogeneous Linear	4.00%	10.33%	12.87%	16.39%	14.57%	13.81%	14.80%
		Heterogeneous Quadratic	4.50%	11.11%	13.00%	15.96%	14.08%	12.88%	13.33%

(Continued)

Table IV. Continued

Data set	User Expectations	Experimental Setting	Recommendation List Size *1*	*3*	*5*	*10*	*30*	*50*	*100*
BookCrossing	Base	Homogeneous Linear	23.08%	9.84%	7.41%	1.90%	2.45%	1.83%	1.02%
		Homogeneous Quadratic	23.08%	10.66%	8.33%	4.05%	3.06%	2.03%	1.23%
		Heterogeneous Linear	12.50%	6.56%	9.26%	4.29%	2.24%	2.43%	1.84%
		Heterogeneous Quadratic	11.54%	6.56%	7.10%	3.57%	1.84%	1.42%	1.02%
	Base+RI	Homogeneous Linear	29.81%	13.52%	8.02%	2.14%	2.65%	2.23%	2.04%
		Homogeneous Quadratic	25.96%	13.52%	8.95%	3.57%	3.67%	2.64%	2.25%
		Heterogeneous Linear	13.46%	7.38%	8.33%	3.10%	2.24%	2.64%	1.64%
		Heterogeneous Quadratic	14.42%	6.56%	7.10%	3.33%	1.63%	1.22%	0.82%
	Base+AR	Homogeneous Linear	22.12%	6.15%	4.32%	−0.48%	1.02%	0.81%	1.02%
		Homogeneous Quadratic	22.12%	7.38%	5.56%	1.19%	1.84%	1.22%	1.23%
		Heterogeneous Linear	8.65%	2.05%	4.63%	0.71%	0.20%	0.81%	0.20%
		Heterogeneous Quadratic	12.50%	5.74%	6.17%	2.86%	1.02%	1.01%	0.61%

Note: Recommendation lists of size $k \in \{20, 40, 60, 70, 80, 90\}$ were not included because of space limitations.

D. CATALOG COVERAGE AND AGGREGATE RECOMMENDATION DIVERSITY

Table V. Increase in Coverage Performance for the MovieLens and BookCrossing Datasets

Data set	User Expectations	Experimental Setting	Recommendation List Size *1*	*3*	*5*	*10*	*30*	*50*	*100*
MovieLens	Base	Homogeneous Linear	38.58%	37.05%	35.15%	28.35%	16.27%	12.38%	7.70%
		Homogeneous Quadratic	38.41%	36.48%	34.65%	28.32%	16.62%	12.47%	7.77%
		Heterogeneous Linear	58.33%	56.29%	55.56%	48.75%	34.71%	30.49%	27.12%
		Heterogeneous Quadratic	52.64%	50.99%	49.55%	42.21%	28.15%	23.38%	19.12%
	Base+RL	Homogeneous Linear	40.00%	37.41%	35.91%	28.93%	16.88%	13.11%	8.82%
		Homogeneous Quadratic	39.41%	37.01%	35.28%	28.65%	17.04%	13.32%	9.38%
		Heterogeneous Linear	63.43%	62.77%	61.29%	53.80%	39.09%	34.62%	30.60%
		Heterogeneous Quadratic	59.16%	57.61%	56.31%	48.77%	34.67%	29.81%	25.71%
BookCrossing	Base	Homogeneous Linear	46.55%	30.27%	21.69%	12.84%	5.66%	4.09%	2.97%
		Homogeneous Quadratic	46.16%	29.79%	21.33%	12.72%	5.56%	4.06%	2.90%
		Heterogeneous Linear	56.77%	40.50%	31.45%	22.71%	16.96%	17.67%	20.31%
		Heterogeneous Quadratic	52.54%	35.67%	26.34%	16.54%	8.68%	7.68%	7.78%
	Base+RI	Homogeneous Linear	36.60%	23.92%	17.31%	10.84%	5.19%	4.67%	5.52%
		Homogeneous Quadratic	35.42%	22.78%	16.15%	9.43%	3.51%	2.94%	4.24%
		Heterogeneous Linear	65.11%	48.12%	38.85%	29.81%	22.75%	22.11%	22.20%
		Heterogeneous Quadratic	60.61%	43.07%	33.55%	23.63%	15.32%	13.92%	14.34%
	Base+AR	Homogeneous Linear	35.26%	21.74%	15.19%	8.80%	2.84%	1.97%	1.36%
		Homogeneous Quadratic	34.04%	20.43%	13.86%	7.31%	0.76%	−0.48%	−1.59%
		Heterogeneous Linear	63.52%	46.43%	37.12%	27.70%	20.53%	19.96%	20.29%
		Heterogeneous Quadratic	59.19%	41.13%	31.52%	21.35%	12.26%	10.47%	9.62%

Note: Recommendation lists of size $k \in \{20, 40, 60, 70, 80, 90\}$ were not included because of space limitations.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

REFERENCES

Panagiotis Adamopoulos. 2013a. Beyond rating prediction accuracy: On new perspectives in recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13)*. ACM, New York, NY, 4.

Panagiotis Adamopoulos. 2013b. What Makes a Great MOOC? An interdisciplinary analysis of online course student retention. In *Proceedings of the 34th International Conference on Information Systems (ICIS'13)*.

Panagiotis Adamopoulos and Alexander Tuzhilin. 2011. On unexpectedness in recommender systems: Or how to expect the unexpected. In *DiveRS 2011 ACM RecSys 2011 Workshop on Novelty and Diversity in Recommender Systems (RecSys'11)*. ACM, New York, NY.

Panagiotis Adamopoulos and Alexander Tuzhilin. 2013a. *Probabilistic Neighborhood Selection in Collaborative Filtering Systems*. *Working Paper CBA-13-04, New York University.* Retrieved from http://hdl.handle.net/2451/31988.

Panagiotis Adamopoulos and Alexander Tuzhilin. 2013b. Recommendation opportunities: Improving item prediction using weighted percentile methods in collaborative filtering systems. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13)*. ACM, New York, NY, 4.

Panagiotis Adamopoulos. 2014. On discovering non-obvious recommendations: Using unexpectedness and neighborhood selection methods in collaborative filtering systems. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM'14)*. ACM, New York, NY, 655–660. DOI:http://doi.acm.org/10.1145/2556195.2556204

Panagiotis Adamopoulos and Alexander Tuzhilin. 2014. On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14)*. ACM, New York, NY, 153–160. DOI:http://doi.acm.org/10.1145/2645710.2645752

G. Adomavicius and Y. Kwon. 2009. Toward more diverse recommendations: Item re-ranking methods for recommender dystems. In *Proceedings of the 19th Workshop on Information Technology and Systems (WITS'09)*.

G. Adomavicius and YoungOk Kwon. 2011. Maximizing aggregate recommendation diversity: A graph-theoretic approach. In *DiveRS 2011 ACM RecSys 2011 Workshop on Novelty and Diversity in Recommender Systems (RecSys'11)*. ACM, New York, NY.

G. Adomavicius and YoungOk Kwon. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (May 2012), 896–911. DOI:http://dx.doi.org/10.1109/TKDE.2011.15

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 (June 2005), 734–749. DOI:http://dx.doi.org/10.1109/TKDE.2005.99

Gediminas Adomavicius and Jingjing Zhang. 2012. Impact of data characteristics on recommender systems performance. *ACM Transactions on Management of Information Systems* 3, 1, Article 3 (April 2012), 17 pages. DOI:http://dx.doi.org/10.1145/2151163.2151166

Takayuki Akiyama, Kiyohiro Obara, and Masaaki Tanizaki. 2010. Proposal and evaluation of serendipitous recommendation method using general unexpectedness. In *Proceedings of the ACM RecSys Workshop on Practical Use of Recommender Systems, Algorithms and Technologies (PRSAT 2010) (RecSys 2010)*. ACM, New York, NY, USA. http://ir.ii.uam.es/prsat2010/papers/paper1.pdf.

Amazon 2012. Amazon.com, Inc. Retrieved from http://www.amazon.com.

Paul André, Jaime Teevan, and Susan T. Dumais. 2009. From x-rays to Silly Putty via Uranus: Serendipity and its role in web search. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*. ACM, New York, NY, 2033–2036. DOI:http://dx.doi.org/10.1145/1518701.1519009

J. Yannis Bakos. 1997. Reducing buyer search costs: Implications for electronic marketplaces. *Management Science* 43, 12 (Dec. 1997), 1676–1692. DOI:http://dx.doi.org/10.1287/mnsc.43.12.1676

William J. Baumol and Edward A. Ide. 1956. Variety in retailing. *Management Science* 3, 1 (1956), 93–101. http://www.jstor.org/stable/2627176.

Robert M. Bell, Jim Bennett, Yehuda Koren, and Chris Volinsky. 2009. The million dollar programming prize. *IEEE Spectrum* 46, 5 (May 2009), 28–33. DOI:http://dx.doi.org/10.1109/MSPEC.2009.4907383

Gideon Berger and Alexander Tuzhilin. 1998. Discovering unexpected patterns in temporal data using temporal logic. In *Temporal Databases: Research and Practice.* Etzion O., Jajodia S., and Sripada S. (Eds.), Vol. 1399, Springer-Verlag Berlin Heidelberg, 281–309.

Michael J. Berry and Gordon Linoff. 1997. *Data Mining Techniques: For Marketing, Sales, and Customer Support*. John Wiley & Sons, New York, NY.

Daniel Billsus and Michael J. Pazzani. 2000. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction* 10, 2–3 (February 2000), 147–180. DOI:http://dx.doi.org/10.1023/A:1026501525781

BookCrossing. 2004. BookCrossing, Inc. Retrieved from http://www.bookcrossing.com.

Erik Brynjolfsson, Yu (Jeffrey) Hu, and Duncan Simester. 2011. Goodbye Pareto Principle, hello long tail: The effect of search costs on the concentration of product sales. *Management Science* 57, 8 (August 2011), 1373–1386. DOI:http://dx.doi.org/10.1287/mnsc.1110.1371

Erik Brynjolfsson, Yu (Jeffrey) Hu, and Michael D. Smith. 2003. Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers. *Management Science* 49, 11 (November 2003), 1580–1596. DOI:http://dx.doi.org/10.1287/mnsc.49.11.1580.20580

Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12, 4 (November 2002), 331–370. DOI:http://dx.doi.org/10.1023/A:1021240730564

Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. 2nd workshop on information heterogeneity and fusion in recommender systems (HetRec 2011). In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys'11)*. ACM, New York, NY.

P. Castells, S. Vargas, and J. Wang. 2011. Novelty and diversity metrics for recommender dystems: Choice, discovery and relevance. In *International Workshop on Diversity in Document Retrieval (DDR'11) at the 33rd European Conference on Information Retrieval (ECIR'11)*.

Òscar Celma and Perfecto Herrera. 2008. A new approach to evaluating novel recommendations. In *Proceedings of the 2nd ACM Conference on Recommender Systems (RecSys'08)*. ACM, New York, NY, 179–186. DOI:http://dx.doi.org/10.1145/1454008.1454038

Helmuth Cremer and Jacques-Francois Thisse. 1991. Location models of horizontal differentiation: A special case of vertical differentiation models. *Journal of Industrial Economics* 39, 4 (1991), 383–390.

Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Vittorio Papadopoulos, and Roberto Turrin. 2011. Looking for "good" recommendations: A comparative evaluation of recommender systems. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-Computer Interaction - Volume Part III (INTERACT'11)*. Springer-Verlag, Berlin, 152–168.

R. Davidson and J. G. MacKinnon. 2004. *Econometric Theory and Methods*. Oxford University Press.

Daniel Fleder and Kartik Hosanagar. 2009. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management Science* 55, 5 (2009), 697–712.

Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10)*. ACM, New York, NY, 257–260. DOI:http://dx.doi.org/10.1145/1864708.1864761

C. Gini. 1909. Concentration and dependency ratios (in Italian). *English translation in Rivista di Politica Economica* 87 (1909), 769–789.

D. G. Goldstein and D. C. Goldstein. 2006. Profiting from the long tail. *Harvard Business Review* 84, 6 (2006), 24–28.

Google. 2012. Google Books. Retrieved from http://books.google.com.

Michele Gorgoglione, Umberto Panniello, and Alexander Tuzhilin. 2011. The effect of context-aware recommendations on customer purchasing behavior and trust. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys'11)*. ACM, New York, NY, 85–92. DOI:http://dx.doi.org/10.1145/2043932.2043951

W. H. Greene. 2012. *Econometric Analysis*. Prentice Hall.

GroupLens. 2011. GroupLens Research Group. Retrieved from http://www.grouplens.org.

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 (January 2004), 5–53. DOI:http://dx.doi.org/10.1145/963770.963772

Yoshinori Hijikata, Takuya Shimizu, and Shogo Nishida. 2009. Discovery-oriented collaborative filtering for improving user satisfaction. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI'09)*. ACM, New York, NY, 67–76. DOI:http://dx.doi.org/10.1145/1502650.1502663

Edgar Hoover. 1985. *An Introduction to Regional Economics*. A. A. Knopf, New York.

Leo Iaquinta, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, Michele Filannino, and Piero Molino. 2008. Introducing serendipity in a content-based recommender system. In *Proceedings of the 8th International Conference on Hybrid Intelligent Systems (HIS'08)*. IEEE Computer Society, Washington, DC, 168–173. DOI:http://dx.doi.org/10.1109/HIS.2008.25

IMDb. 2011. IMDb.com, Inc. Retrieved from http://www.imdb.com.

ISBNdb.com. 2012. The ISBN Database. Retrieved from http://isbndb.com.

Noriaki Kawamae. 2010. Serendipitous recommendations via innovators. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*. ACM, New York, NY, 218–225. DOI:http://dx.doi.org/10.1145/1835449.1835487

Noriaki Kawamae, Hitoshi Sakano, and Takeshi Yamada. 2009. Personalized recommendation based on the personal innovator degree. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys'09)*. ACM, New York, NY, 329–332. DOI:http://dx.doi.org/10.1145/1639714.1639780

M. Khabbaz, M. Xie, and L. V. S. Lakshmanan. 2011. TopRecs: Pushing the envelope on recommender systems. *Data Engineering* (2011), 61.

F. M. Khan and V. B. Zubek. 2008. Support vector regression for censored data (SVRc): A novel tool for survival analysis. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08)*. 863–868. DOI:http://dx.doi.org/10.1109/ICDM.2008.50

Joseph A. Konstan, Sean M. McNee, Cai-Nicolas Ziegler, Roberto Torres, Nishikant Kapoor, and John T. Riedl. 2006. Lessons on applying automated recommender systems to information-seeking tasks. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2 (AAAI'06)*. AAAI Press, Palo Alto, CA, 1630–1633.

J. A. Konstan and J. T. Riedl. 2012. Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction* 22 (2012), 101–123. DOI:http://dx.doi.org/10.1007/s11257-011-9112-x

Kleanthis-Nikolaos Kontonasios, Eirini Spyropoulou, and Tijl De Bie. 2012. Knowledge discovery interestingness measures based on unexpectedness. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2, 5 (2012), 386–399. DOI:http://dx.doi.org/10.1002/widm.1063

Yehuda Koren. 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery Data* 4, 1 (January 2010), Article 1, 24 pages. DOI:http://dx.doi.org/10.1145/1644873.1644874

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (August 2009), 30–37. DOI:http://dx.doi.org/10.1109/MC.2009.263

Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. 2010. Temporal diversity in recommender systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*. ACM, New York, NY, 210–217. DOI:http://dx.doi.org/10.1145/1835449.1835486

Daniel Lemire and Anna Maclachlan. 2007. Slope one predictors for online rating-based collaborative filtering. *CoRR* abs/cs/0702144 (2007).

LibraryThing. 2012. LibraryThing. Retrieved from http://www.librarything.com.

J. S. Long. 1997. *Regression Models for Categorical and Limited Dependent Variables*. Vol. 7. Sage Publications, Inc.

M. O. Lorenz. 1905. Methods of measuring the concentration of wealth. *Publications of the American Statistical Association* 9, 70 (1905), 209–219.

Alfred Marshall. 1920. *Principles of Economics*. Vol. 1. Macmillan and Co., London, UK.

John F. McDonald and Robert A. Moffitt. 1980. The uses of Tobit analysis. *The Review of Economics and Statistics* 62, 2 (1980), 318–321.

Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems (CHI EA'06)*. ACM, New York, NY, 1097–1101. DOI:http://dx.doi.org/10.1145/1125451.1125659

David McSherry. 2002. Diversity-conscious retrieval. In *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning (ECCBR'02)*. Springer-Verlag, London, UK, 219–233.

Tomoko Murakami, Koichiro Mori, and Ryohei Orihara. 2008. Metrics for evaluating the serendipity of recommendation lists. In *Proceedings of the 2007 Conference on New Frontiers in Artificial Intelligence (JSAI'07)*. Springer-Verlag, Berlin, 40–46.

Makoto Nakatsuji, Yasuhiro Fujiwara, Akimichi Tanaka, Toshio Uchiyama, Ko Fujimura, and Toru Ishida. 2010. Classical music for rock fans?: Novel recommendations for expanding user interests. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*. ACM, New York, NY, 949–958. DOI:http://dx.doi.org/10.1145/1871437.1871558

Netflix. 2012. Netflix, Inc. Retrieved from http://www.netflix.com.

Damien Neven. 1985. Two stage (perfect) equilibrium in Hotelling's model. *The Journal of Industrial Economics* 33, 3 (1985), 317–325.

Randall J. Olsen. 1978. Note on the uniqueness of the maximum likelihood estimator for the Tobit model. *Econometrica* 46, 5 (1978), 1211–1215.

Balaji Padmanabhan and Alexander Tuzhilin. 1998. A belief-driven method for discovering unexpected patterns. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD'98)*. AAAI Press, Palo Alto, CA, 94–100.

Balaji Padmanabhan and Alexander Tuzhilin. 2000. Small is beautiful: Discovering the minimal set of unexpected patterns. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00)*. ACM, New York, NY, 54–63. DOI:http://dx.doi.org/10.1145/347090.347103

Balaji Padmanabhan and Alexander Tuzhilin. 2006. On characterization and discovery of minimal unexpected patterns in rule discovery. *IEEE Transactions on Knowledge and Data Engineering* 18, 2 (February 2006), 202–216. DOI:http://dx.doi.org/10.1109/TKDE.2006.32

Umberto Panniello, Alexander Tuzhilin, Michele Gorgoglione, Cosimo Palmisano, and Anto Pedone. 2009. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys'09)*. ACM, New York, NY, 265–268. DOI:http://dx.doi.org/10.1145/1639714.1639764

S. Rabe-Hesketh, A. Skrondal, and A. Pickles. 2002. Reliable estimation of generalized linear mixed models using adaptive quadrature. *Stata Journal* 2, 1 (2002), 1–21.

Alan Said, Brijnesh J. Jain, Benjamin Kille, and Sahin Albayrak. 2012. Increasing diversity through furthest neighbor-based recommendation. In *Proceedings of the WSDM'12 Workshop on Diversity in Document Retrieval (DDR'12)*.

Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. *Recommender Systems Handbook* 12, 19 (2011), 1–41.

P. K. Shivaswamy, Wei Chu, and M. Jansche. 2007. A support vector approach to censored targets. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM'07)*. 655–660. DOI:http://dx.doi.org/10.1109/ICDM.2007.93

A. Silberschatz and A. Tuzhilin. 1996. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering* 8, 6 (December 1996), 970–974. DOI:http://dx.doi.org/10.1109/69.553165

Kazunari Sugiyama and Min-Yen Kan. 2011. Serendipitous recommendation for scholarly papers considering relations among researchers. In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries (JCDL'11)*. ACM, New York, NY, 307–310. DOI:http://dx.doi.org/10.1145/1998076.1998133

Jean Tirole. 1988. *The Theory of Industrial Organization*. MIT Press.

Akhmed Umyarov and Alexander Tuzhilin. 2011. Using external aggregate ratings for improving individual recommendations. *ACM Transactions on the Web* 5, 1 (February 2011), Article 3, 40 pages. DOI:http://dx.doi.org/10.1145/1921591.1921594

Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys'11)*. ACM, New York, NY, 109–116. DOI:http://dx.doi.org/10.1145/2043932.2043955

Li-Tung Weng, Yue Xu, Yuefeng Li, and Richi Nayak. 2007. Improving recommendation novelty based on topic taxonomy. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops (WI-IATW'07)*. IEEE Computer Society, Washington, DC, 115–118.

Wikipedia. 2012. Wikimedia Foundation, Inc. Retrieved from http://www.wikipedia.org.

J. M. Wooldridge. 2002. *Econometric Analysis of Cross Section and Panel Data*. MIT Press.

WorldCat 2012. OCLC Online Computer Library Center, Inc. Retrieved from http://www.worldcat.org.

Mi Zhang and Neil Hurley. 2008. Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys'08)*. ACM, New York, NY, 123–130. DOI:http://dx.doi.org/10.1145/1454008.1454030

Mi Zhang and Neil Hurley. 2009. Novel item recommendation by user profile partitioning. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01 (WI-IAT'09)*. IEEE Computer Society, Washington, DC, 508–515. DOI:http://dx.doi.org/10.1109/WI-IAT.2009.85

Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: Introducing serendipity into music recommendation. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM'12)*. ACM, New York, NY, 13–22. DOI:http://dx.doi.org/10.1145/2124295.2124300

T. Zhou, Z. Kuscsik, J. G. Liu, M. Medo, J. R. Wakeling, and Y. C. Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4511.

Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web (WWW'05)*. ACM, New York, NY, 22–32. DOI:http://dx.doi.org/10.1145/1060745.1060754

Received September 2012; revised August 2013; accepted November 2013