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 paper, 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 expect from the system. We define and formalize the concept of unexpectedness and discuss how it differs from the related notions of novelty, serendipity, and diversity. Besides, 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 the 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" data sets to compare our recommendation results with some other standard baseline methods. The proposed approach outperforms these baseline methods in terms of unexpectedness and other important metrics, such as coverage and aggregate diversity, while avoiding any accuracy loss.

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

Additional Key Words and Phrases: Evaluation, Novelty, Recommendations, Recommender Systems, Serendipity, Unexpectedness, Utility Theory

"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.

1. INTRODUCTION

Over the last decade, a wide variety of different types of recommender systems (RSs) has been developed and used across several domains [Adomavicius and Tuzhilin 2005]. Although the broad-based social and business acceptance of RSs 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 due, primarily, to the fact that many existing RSs focus on providing more accurate rather than more novel, serendipitous, diverse, and useful recommendations. Some of the main problems pertaining to the narrow accuracy-based focus of many existing RSs and the ways to broaden the current approaches have been discussed in [McNee et al. 2006].

One of the key dimensions for improvement that can significantly contribute to the overall performance and usefulness of RSs, and is still under-explored, is the notion of *unexpectedness*. RSs often recommend expected items that the users are already familiar with and, thus, they are of little interest to them. For example, a shopping RS may recommend to customers products such as milk and bread. Although being accurate, in the sense that the customer will indeed buy these two products,

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

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).

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 the 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 user and then derive the unexpected ones, but also to enhance the concept of unexpectedness while still delivering recommendations of high quality that achieve a fair match to user's interests.

In this paper, we formalize this concept by providing a new formal definition of unexpected recommendations, as those recommendations that significantly depart from user's expectations, and differentiate it from various related concepts, such as novelty and serendipity. We also propose a 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 particular, *novel* recommendations are recommendations of those items that the user did not know about [Konstan et al. 2006]. Hijikata et al. [2009] use collaborative filtering to derive novel recommendations by explicitly asking users what items they already know. Besides, [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 compose the recommendation list 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 and [Nakatsuji et al. 2010] define and measure novelty as the smallest distance from the class the user accessed before to the class that includes target items over the taxonomy. 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.

Moreover, *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. [Iaquinta et al. 2008] propose to enhance serendipity by recommending novel items whose description is semantically far from users' profiles and [Kawamae et al. 2009], [Kawamae 2010] suggest an algorithm for recommending novel items based on the assumption that users follow earlier adopters who have demonstrated similar preferences. In addition, [Sugiyama and Kan 2011] proposed 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 of 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 constitutes a different type of recommendations that can be more risky and ambiguous. To further illustrate the differences of these two concepts, let's 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 (non-novel) Action film of another director. The user will most probably like this unexpected but non-serendipitous recommendation.

Furthermore, *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 to minimize similarity between all items or remove "obvious" items from them as in [Billsus and Pazzani 2000]. [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 list. Then, [Zhang and Hurley 2008] focus on intra-list diversity and address the problem as the joint optimization of two objective functions reflecting preference similarity and item diversity, and [Hurley and Zhang 2011] formulate the trade-off between diversity and matching quality as a binary optimization problem. Besides, [Wang and Zhu 2009], inspired by the modern portfolio theory in financial markets, suggest an algorithm that generalizes the probability ranking principle by considering both the uncertainty of relevance predictions and correlations between retrieved documents. Also, [Said et al. 2012] suggest an inverted nearest neighbor model and recommend items disliked by the least similar users. 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 which is designed to deliver such similarity-preserving increases in diversity. In addition, [Zhang et al. 2012] propose a collection of algorithms to simultaneously increase novelty, diversity, and serendipity, at a slight cost to accuracy, and [Zhou et al. 2010] suggest a hybrid algorithm which, without relying on any semantic or context-specific information, simultaneously gains in both accuracy and diversity of recommendations. In another stream of research, [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 RS literature [Zhang and Hurley 2008], [Zhou et al. 2010], [Ziegler et al. 2005], and derive new ones. They suggest novelty and diversity metric schemes that take into consideration item position and relevance through a probabilistic recommendation browsing model. Besides, other researchers studied the importance of personalization and users' perception in diversity. In particular, [Hu and Pu 2011] investigate design issues that can enhance users' perception or recommendation diversity and improve users' satisfaction, and [Ge et al. 2012] show that the perceived diversity of a recommendation list depends on the placement of diverse items. Further, [Vargas et al. 2012] suggest that the combination of personalization and diversification achieves competitive performance improving the baseline, plain personalization, and plain diversification approaches in terms of both diversity and accuracy measures, and [Shi et al. 2012] argue that the diversification level in a recommendation list should be adapted to the target users' individual situations and needs, and propose a framework to adaptively diversify recommendation results for individual users based on latent factor models. Lastly, examining similar but yet different concepts of diversity, [Adomavicius and Kwon 2009; 2012] propose the concept of aggregated 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. Also, [Lathia et al. 2010] consider the concept of temporal diversity, the 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 the 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.

Pertaining to *unexpectedness*, in the field of knowledge discovery, [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, which combine the independent concepts of unexpectedness and minimality of patterns. Also, [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] suggest both a definition of unexpectedness as the deviation from the results obtained from a primitive prediction model and metrics for evaluating unexpectedness. Also, [Akiyama et al. 2010] propose unexpectedness as a general metric that does not depend on a user's record and involves an unlikely combination of features. However, all these approaches do not fully capture the multi-faceted 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 definition of unexpectedness, taking into account prior expectations of the user, and methods for providing unexpected recommendations are still needed. In this paper, we deviate from the previous definitions of unexpectedness and propose a new formal definition that we present in the next section.

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 for providing truly useful recommendations since it is possible to deliver unexpected recommendations but of low quality. Therefore, after defining unexpectedness, we introduce *utility* of a recommendation and provide an example of utility as a function of the *quality* of 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 measures for evaluating the generated recommendations. We define unexpectedness in Section 3.1, 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 start with user expectations. The *expected items* for each user u can be defined as a finite collection of items that the user considers as choice candidates in order to serve her own current needs or fulfil her intentions, as indicated by interacting with the recommender system. This 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 expects to receive. The sets of user expectations, as the true expectations of the users, can also be adapted to different contexts and evolve with the time. For example, in case of a movie RS, this set of expected items may include all the movies seen by the user *and* all their related and similar movies, where "relatedness" and "similarity" are defined 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, *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 prefered 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 though 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.

The above definitions[1] clearly take into consideration the actual *expectations of the 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 one. It is the interaction of the user and the item that characterizes whether the particular recommendation is unexpected for the specific user or not.

However, recommending to a user the items that result in the highest level of unexpectedness would be problematic sometimes, since recommendations should also be of high quality and fairly match users' preferences. In other words, it is important to highlight that simply increasing the unexpectedness of a recommendation list is worthless if this list does not contain relevant items of high quality that the user likes. In order to generate such recommendations that would maximize the users' satisfaction, we use certain concepts from the utility theory in economics [Marshall 1920].

[1] The aforementioned definitions serve as templates that are precisely defined and operationalized through specific mechanisms in Sections 4.2.1-4.2.4.

3.2 Utility of Recommendations

In the context of recommender systems, pertaining to the concept of unexpectedness and 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 using the product and the utility of unexpectedness of the recommended item, as defined in Section 3.1. Our proposed model follows the standard assumption in economics that the users are engaging into 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 and hence, given the unexpectedness of an item, the greater the quality of this item, the greater the utility of the recommendation to the user. Consequently, without loss of generality, we propose that 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 will allow the utility function to have the required characteristics described so far. Note that the distribution of utility as a function of unexpectedness and quality is non-linear, bounded, and experiences a global maximum.

Formalizing these concepts, in order 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 the 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 of the user decreases by departing from the preferred level of unexpectedness δ_u^*. Then, the utility of the 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 for user u and item i.

Then, utility of recommending items to users 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 is linearly increasing with the rating for these distances, whereas, given the quality of the product, it increases with unexpectedness up to the threshold of

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 unexpectedness of item $i$ 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
```

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 $U_{u,i}$ 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 restrictions can be applied on the quality and unexpectedness of the candidate items, if appropriate in the application, in order 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 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 estimate the unexpectedness of the specific item for the particular user $\delta_{u,i}$. Then, if the estimated unexpectedness is within the specified interval $[\underline{\delta}, \bar{\delta}]$, we estimate the utility of recommending this item to the specific user $U_{u,i}$. Finally, we recommend to the user the items that exhibit the highest estimated utility. Examples on how to compute the set of expected item E_u for a user are provided in Section 4.2.3.

[2] Eqs. (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). Besides, function ϕ does not have to be symmetric as in the examples provided in (7) and (8).

[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 tuning of its coefficients.

3.4 Evaluation of Recommendations

[Adomavicius and Tuzhilin 2005], [Herlocker et al. 2004], [McNee et al. 2006] suggest that RS 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 generated recommendations.

3.4.1 *Metrics of Unexpectedness.* In order to measure unexpectedness, we follow the approach proposed by [Murakami et al. 2008] and [Ge et al. 2010], and adapt their measures to our method. In particular, Ge et al. [2010] define an unexpected set of recommendations (UNEXP) as:

$$\mathrm{UNEXP} = \mathrm{RS} \setminus \mathrm{PM} \tag{9}$$

where PM is a set of recommendations generated by a primitive prediction model, such as predicting items based on users' favorite categories or items' number of ratings, 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] argues, unexpected recommendations may not be always useful and, thus, the paper also introduces serendipity measure as:

$$\mathrm{SRDP} = \frac{|\mathrm{UNEXP} \bigcap \mathrm{USEFUL}|}{|N|} \tag{10}$$

where USEFUL denotes the set of "useful" items and N the length of the recommendation list. For instance, the usefulness of an item can be judged by the users or approximated by the items' ratings as we describe in Section 4.2.6.

However, these measures do not fully capture the proposed definition of unexpectedness since PM contains the most popular items and does not actually take into account *the expectations of the user*. Consequently, we revise their definition and introduce new metrics to measure unexpectedness as follows. First of all, we define expectedness (EXPECTED) as the mean ratio of the items which are included in both the set of expected recommendations for a user (E_u) and the generated recommendation list (RS_u):

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

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

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

Correspondingly, we can also derive a new metric for serendipity as in (10) based on the proposed metric of unexpectedness:

$$\mathrm{SERENDIPITY} = \sum_u \frac{|(\mathrm{RS}_u \setminus \mathrm{E}_u) \bigcap \mathrm{USEFUL}_u|}{|N|}. \tag{13}$$

For the sake of simplicity, the metrics defined so far consider whether an item is expected to the user or not 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. Besides, 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].

3.4.2 *Metrics of Accuracy*. The recommendation lists can also be evaluated for the accuracy of rating and item predictions using standard measures such as Root Mean Square Error, Mean Absolute Error, Precision, Recall, and F-measure. In applications where the number of recommendations presented to the user is preordained, the most useful measure of interest is precision at N [Shani and Gunawardana 2011].

Finally, recommender systems can also be evaluated based on various other of metrics including diversity, confidence, trust, robustness, adaptivity, and catalog coverage [Shani and Gunawardana 2011].

4. EXPERIMENTAL SETTINGS

To empirically validate the method presented in Section 3.3 and evaluate the unexpectedness of the generated recommendations, we conduct a large number of experiments on "real-world" data sets and compare our results to popular baseline methods.

Unfortunately, we could not compare our results with other methods for deriving unexpected recommendations for the following reasons. First, most of the existing methods are based on related but different principles such as diversity and novelty. Since these concepts are, in principle, different from our definition, they cannot be directly compared with our approach. Further, among the previously proposed methods of unexpectedness that are consistent with our approach, as explained in Section 2, the authors present only the performance metrics and do not provide any clear computational algorithm for computing recommendations, thus making the comparison impossible. Consequently, we selected a number of standard Collaborative Filtering (CF) and other 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 constitute fairly reasonable baselines because, as was pointed out in [Burke 2002], CF methods also perform well in terms of other performance measures besides the classical accuracy measures.

4.1 Data sets

The basic data sets that we used are the RecSys HetRec 2011 MovieLens data set [Cantador et al. 2011] and the BookCrossing data set [Ziegler et al. 2005].

The RecSys HetRec 2011 MovieLens (ML) data set [2011] is an extension of a data set published by [GroupLens 2011], which contains personal ratings and tags about movies, and consists of 855,598 ratings from 2,113 users on 10,197 movies. This data set is relatively dense (3.97%) compared to other frequently used data sets 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. Besides, in order to test the proposed method under various levels of sparsity [Adomavicius and Zhang 2012], we consider different proper subsets of the data sets.

In addition, we used information and further details from Wikipedia [Wikipedia 2012] and IMDb [IMDb 2011]. Joining these data sets we were able to enhance the available information by identifying whether a movie is an episode or sequel of another movie included in our data set. 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 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 MovieLens data set consisting of 4,735 items and 2,029 users, with at least 25 ratings each, exhibiting 807,167 ratings.

The BookCrossing (BC) data set is gathered by [Ziegler et al. 2005] from Bookcrossing.com [BookCrossing 2004], a social networking site founded to encourage the exchange of books. This data set 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 data set was selected because we can use the implicit ratings of the users to better specify their expectations, as described in Section 4.2.3. Besides, we supplemented the available data for 261,229 books with information from Amazon [Amazon 2012], Google Books [Google 2012], ISBNdb [ISBNdb.com 2012], LibraryThing [LibraryThing 2012], Wikipedia [Wikipedia 2012], and WorldCat [WorldCat 2012]. Such data is often publicly available and, therefore, it 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 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 BookCrossing data set's extreme sparsity, we decided to further condense the set in order to obtain more meaningful results from collaborative filtering 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 4 ratings and community members with fewer than 8 ratings each. The dimensions of the resulting data set were considerably more moderate, featuring 8,824 users, 18,607 books, and 377,749 ratings (147,403 explicit ratings). Finally, we also consider two proper subsets of this; (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 MovieLens data set, 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 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 ($k \in \{1, 3, 5, 10, 20, \dots, 100\}$). In conclusion, we used 2 subsets, 2 sets of expected movies, 6 algorithms for rating prediction, 3 correlation metrics, 2 distance metrics, 2 utility functions, 2 different assumptions about users preferences, and 13 different lengths of recommendation lists, resulting in 7,488 experiments in total.

Using the BookCrossing data set, we conducted our experiments on three different proper subsets described in Section 4.1. As before, we also assume different specifications for experiments. In particular, for each subset, we used 3 subsets, 3 sets of expected books, 6 algorithms for rating prediction, 3 correlation metrics, 2 distance metrics, 2 utility functions, 2 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 (6):

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

(1b) *Representative agent (homogeneous users) with quadratic distance* (Hom-Quad): The users are be homogeneous but $\phi(\delta_{u,i}; \delta^*_u)$ is quadratic in $\delta_{u,i}$ in (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. $q_u, \lambda_u, \delta^*_u$), and $\phi(\delta_{u,i}; \delta^*_u)$ is linear in $\delta_{u,i}$ as in (7):

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

(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 function (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.[4] The distance matrix can be easily updated with respect to new ratings as in [Khabbaz et al. 2011] in order 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 ($\underline{q}$). Other techniques that should also be explored in future research include user clustering, low rank approximation of 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 specified using various mechanisms that can be applied across domains. Such mechanisms are the past transactions performed by the user, knowledge discovery and data mining techniques, and experts' domain knowledge. In order to test the proposed method under various sets of expected recommendations of different cardinalities that have been specified using the mechanisms summarized in Table I, we consider the following settings.

(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/he has already rated. In particular, movie i is expected for user u if the user has already rated some movie j such

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

Table I. : Sets of expected recommendations for different experimental settings.

Data set	**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

that i 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 ($\mathrm{E}_u^{(Base+RL)} \supseteq \mathrm{E}_u^{(Base)}$). In order 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 of them (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 twenty 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 BookCrossing data set, we use three different examples of expected books for our users. The first set of expectations ($\mathrm{E}_u^{(Base)}$) consists of only the items that user u rated implicitly or explicitly.[5] 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 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

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

items included in set E_u Additionally, we also use the Centroid distance that is defined as the distance of an item i from the centroid point of the set of expected recommendations E_u for user u.[6]

4.2.5 *Utility Estimation.* Since the users are restricted to provide ratings on a specific scale, the corresponding item ratings in our data sets are censored from below and above (also known as censoring from left and right, respectively) [Davidson and MacKinnon 2004]. Hence, in order to model the 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 that was available to the users, we borrow from economics popular models of censored multiple linear regressions [McDonald and Moffitt 1980], [Olsen 1978], [Long 1997][7] imposing also a restriction on these models for non-negative coefficients (i.e. $q_u, \lambda_u \geq 0$) [Greene 2012], [Wooldridge 2002].

Furthermore, given the limitations of offline experiments and our data sets, 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, will allow us to study the effect of taking unexpectedness into consideration, without introducing any other source of variation into our model. We also used the average distance of rated items from the set of expected recommendations in order 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. Besides, we used a holdout validation scheme in all of our experiments with 80/20 splits of data to the training/test part in order to avoid overfitting. Finally, we assume an application scenario where an item can be a candidate recommendation for a user if and only if it has not been rated by the specific user; 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 3.4.1. Additionally, we further evaluate the recommendation lists 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 (9) we used the top-N items with highest average rating and the largest number of ratings. For instance, for the experiments conducted using the main subset of the MovieLens data set, the PM model consists of the top 200 items with the highest average rating and top 800 items with the greatest number of ratings; the same ratio was used for all the experiments.

Additionally, we introduce expectedness' (EXPECTED') 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 are also included in the generated recommendation list. Correspondingly, we define unexpectedness' (UNEXPECTED') as the mean ratio of the recommended items that are neither included in expectations nor in the primitive prediction model, and are included in the generated recommendations:

$$\text{UNEXPECTED'} = \sum_u \frac{|\mathrm{RS}_u \setminus (\mathrm{E}_u \cup \mathrm{PM})|}{|N|}. \tag{19}$$

Based on the ratio of Ge et al. [2010] in (10), we also use the metrics SERENDIPITY and SERENDIPITY' to evaluate serendipitous recommendations in conjunction with the metrics of unexpectedness in

[6] The experiments conducted in [Adamopoulos and Tuzhilin 2011] using the Hausdorff distance ($d(i, \mathrm{E}_u) = \inf\{d(i,j) : j \in \mathrm{E}_u\}$) indicate inconsistent performance and sometimes under-performed the standard CF methods. Hence, in this work we only conducted experiments using the average and the centroid distance.

[7] Multiple linear regression 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], [Khan and Zubek 2008] may also be used for utility estimation.

(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, in the experiments conducted using the ML and BC data sets, we consider an item to be useful if its average rating is greater than 2.5 (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, and accuracy of rating and item predictions using the metrics discussed in Section 3.4.

5. RESULTS

The aim of this study is to demonstrate that the proposed method is indeed effectively capturing the concept of unexpectedness and performs well in terms of the 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 (5 subsets based on 2 data sets, 5 sets of expected items, 6 algorithms for rating prediction, 3 correlation metrics, 2 distance metrics, 2 utility functions, 2 different assumptions about users preferences, 13 different lengths of recommendation lists), the total number of the conducted experiments was 24,336, which constitutes a challenging problem to present the results. 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). 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 MovieLens data set. 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 BookCrossing data set, 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.

5.1 Comparison of Unexpectedness

In this section, we experimentally demonstrate that the proposed method effectively captures the notion of unexpectedness and, hence, outperforms the standard baseline methods in terms of unexpectedness. Tables VI and VIII in the Appendix present the results obtained by applying our method to the MovieLens (ML) and BookCrossing (BC) data sets. The values reported are computed using the proposed unexpectedness metric (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. Besides,

Table II. : Unexpectedness Performance for the MovieLens and BookCrossing Data Sets.

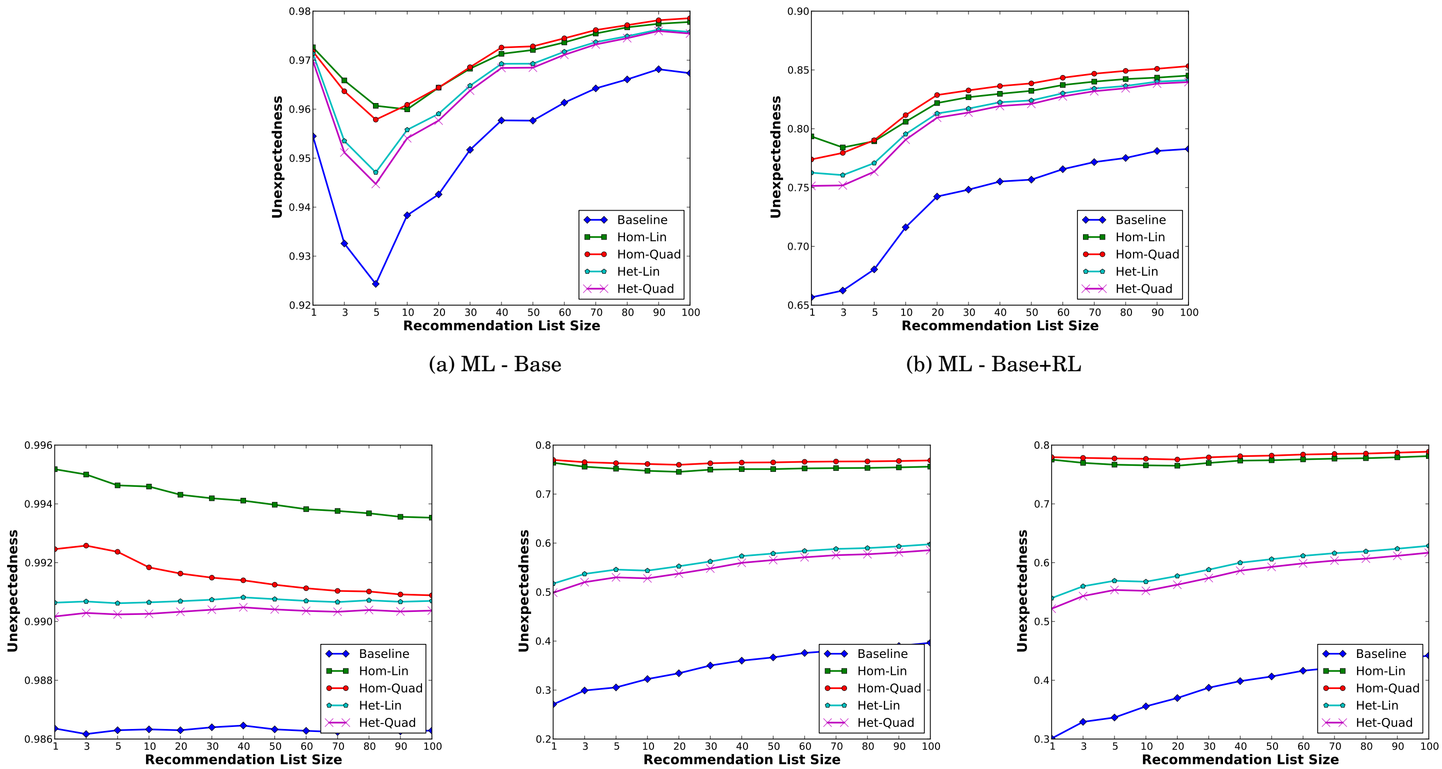
Data	**User**	**Experimental**	**Recommendation List Size**						
Set	**Expectations**	**Setting**	*1*	*3*	*5*	*10*	*30*	*50*	*100*
MovieLens	Base	Homogeneous Linear	1.90%	3.57%	3.93%	2.30%	1.74%	1.51%	1.08%
		Homogeneous Quadratic	1.81%	3.33%	3.63%	2.40%	1.77%	1.58%	1.16%
		Heterogeneous Linear	1.77%	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.

Fig. 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.

Table II and Fig. 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. Fig. 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 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). 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). Besides, Tables VI and VIII indicate that the increase in performance is larger also in the 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 MovieLens data set, 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 BookCrossing data set, 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 MovieLens and BookCrossing data sets, respectively.

A particularly noteworthy observation, as demonstrated through the distribution of unexpectedness for the ML and BC data sets in Fig. 2, is that the higher the cardinality and the better approximated



(a) ML - Base

(b) ML - Base+RL

(c) BC - Base

(d) BC - Base+RI

(e) BC - Base+AR

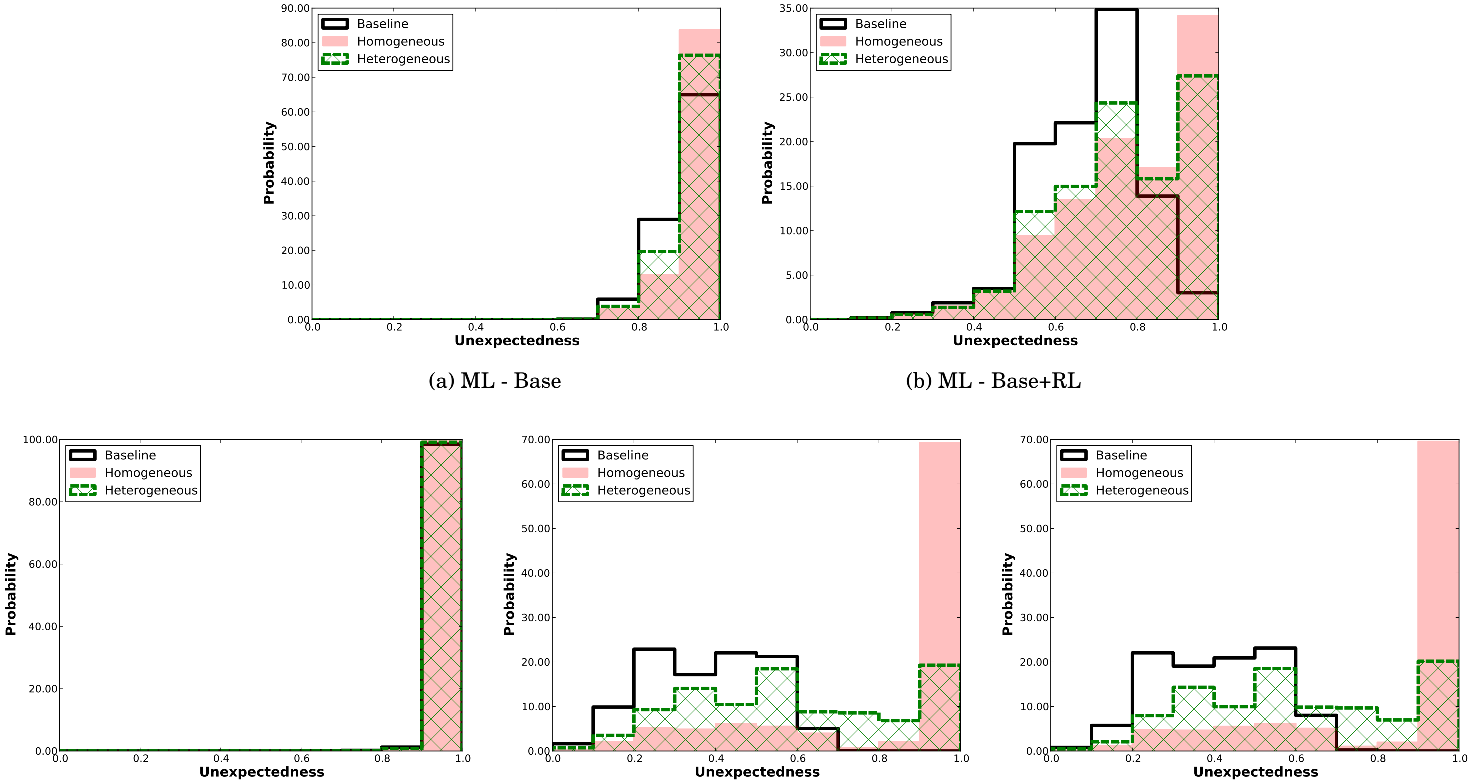
Fig. 1: Unexpectedness performance of different experimental settings for the (a), (b) MovieLens (ML) and (c), (d), (e) BookCrossing (BC) data sets.

the sets of users' expectations are, the greater the improvements against the baseline methods.[8] 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 Fig. 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 line fitting the data for recommendation lists of size $k = 5$.[9] 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 have tested the null hypothesis that the performance of each of the five lines of the graphs in Fig. 1 is the same, using the Friedman test (nonparametric repeated measure ANOVA) [Berry and Linoff 1997] and we reject the null hypothesis with $p < 0.0001$.

[8] Figs. 12 and 13 in the Appendix present the distribution of unexpectedness across all the users for the different rating estimation algorithms using the MovieLens and BookCrossing data sets with the respective sets of user expectations (Base+RL) and (Base+RI), and recommendation lists of size $k = 5$.

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



(a) ML - Base

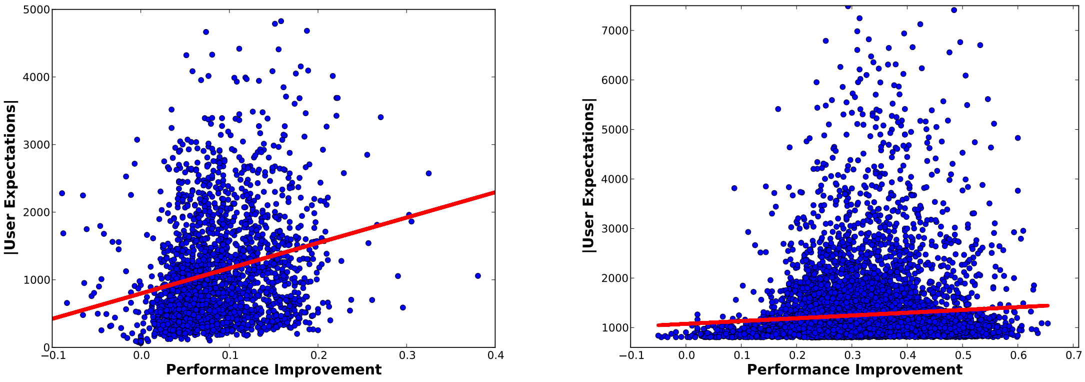
(b) ML - Base+RL

(c) BC - Base

(d) BC - Base+RI

(e) BC - Base+AR

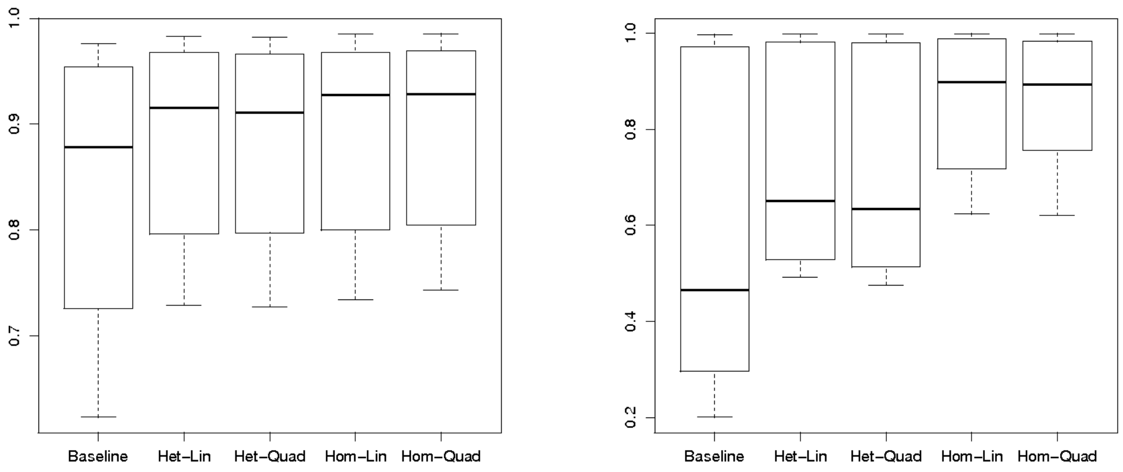
Fig. 2: Distribution of Unexpectedness for recommendation lists of size k=5 and different experimental settings for the MovieLens (ML) and BookCrossing (BC) data sets.



(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) data sets using different sets of expectations.



(a) MovieLens data set

(b) BookCrossing data set

Fig. 4: Post hoc analysis for Friedman's Test of Unexpectedness Performance of different methods for the (a) MovieLens and (b) BookCrossing data sets.

Performing post hoc analysis on Friedman's Test results for the ML data set, 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. Besides, 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 data set, 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. Fig. 4 presents the box-and-whisker diagrams [Benjamini 1988] displaying the aforementioned differences among the various methods.

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" because the user had already highly rated all the sequels or prequels of these movies. Besides, 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.

Using the MovieLens data set 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 Matrix Factorization). 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”) and, hence, 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 the user expectations as described in Section 4.2.3, all these interesting movies are unexpected for the user since they significantly depart from her/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. Besides, 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 which she/he has already highly rated such as “Erin Brockovich”, “October Sky”, and “Three Kings”. Finally, interestingly enough, some of these interesting and unexpected recommendations are also based on movies filmed by the same director that adapted a film the user rated highly (i.e. “Pinocchio” and “La vita è bella”).

Using the BookCrossing data set 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 Matrix Factorization). In particular, the book “I Know This Much Is True” is highly expected because the specific user has already rated and she/he 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, which 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”), etc. In spite of that, the proposed algorithm recommends to the user the following books that significantly depart from her/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, they are not irrelevant since they provide a fair match to user’s interests since she/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, Tables VII and IX in the Appendix present the results obtained by applying our method to the MovieLens and BookCrossing data sets. The values reported are computed using the adapted metric (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 $k \in \{1, 3, 5, 10, 30, 50, 100\}$. Fig. 14 presents the average performance 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. Finally, Fig. 15 presents the box-and-whisker diagrams displaying the statistically significant differences among the various methods. The results are very similar to those obtained using the proposed measures of unexpectedness and demonstrate that the proposed method outperforms the standard baselines in most of the experimental settings.

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, hence, 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 the classical rating prediction accuracy-based metrics, such as RMSE and MAE. In typical offline experiments as those presented here, the data is 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.[10] Also, the user ratings had been submitted over a long period of time representing the tastes of the users and their expectations of the recommender system at that specific point in time that they rated each item. Therefore, in order 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 $\mathrm{E}_u^{-i} := \mathrm{E}_u \setminus \{i\}$.

Tables X - XIII in the Appendix present the results obtained by applying our method to the ML and BC data sets. 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 Fig. 5, the bars labeled as *Baseline* represent performance of the standard baseline methods. The bars labeled as *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 data set the RMSE was on average reduced by 0.07% and 0.34% for the cases of the homogeneous and heterogeneous users. For the BC data set, 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 data set.

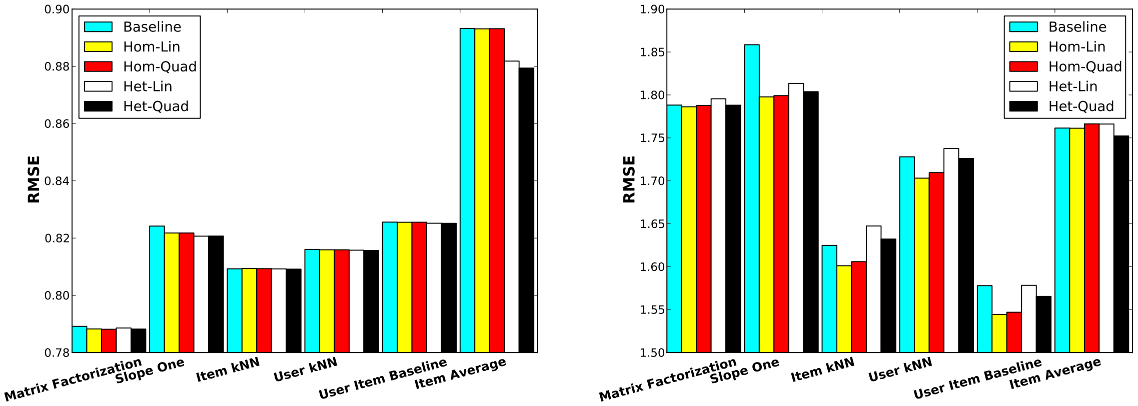
Using the Friedman test, we have tested the null hypothesis that the performance of each of the five lines of the graphs in Fig. 5 is the same; we reject the null hypothesis with $p < 0.001$. Performing post hoc analysis on Friedman's Test results, for the ML data set only the difference between the *Heterogeneous Quadratic* and *Baseline* is statistically significant for the RMSE accuracy metric. For the BC data set, the differences between the *Homogeneous Linear* and *Baseline*, and *Homogeneous Quadratic* and *Baseline* are statistically significant, as well. Fig. 6 presents the box-and-whisker diagrams displaying the aforementioned differences among the various methods.

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.

[10] 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/he was unaware of its existence, but after the recommendation exposed that item the user can decide to select it [Shani and Gunawardana 2011]).

Table III. : Average RMSE Performance for the MovieLens and BookCrossing Data Sets.

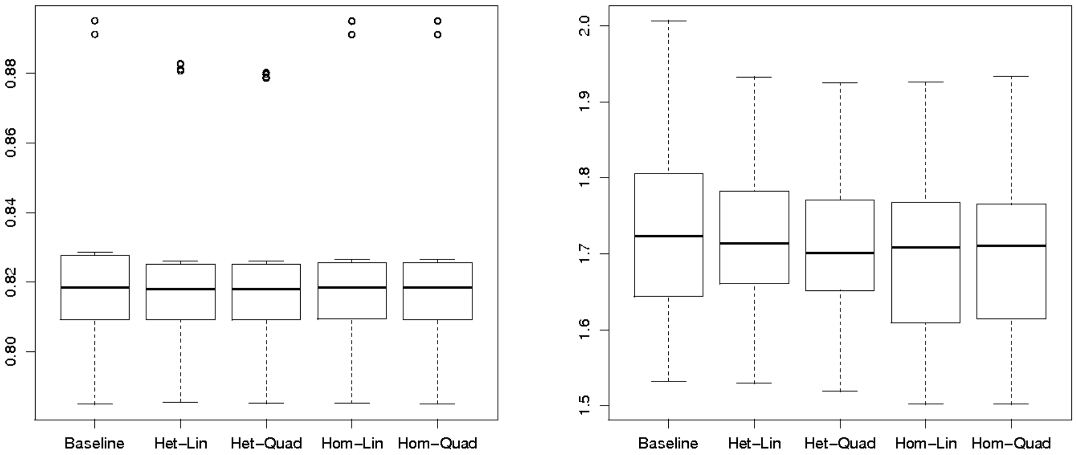
Data Set	Rating Prediction Algorithm	Expectations	Baseline	Homogeneous		Heterogeneous	
				Linear	Quadratic	Linear	Quadratic
MovieLens	MatrixFactorization	Base	0.7892	0.11%	0.13%	0.07%	0.12%
		Base+RL	0.7892	0.12%	0.13%	0.07%	0.12%
	SlopeOne	Base	0.8242	0.29%	0.29%	0.43%	0.43%
		Base+RL	0.8242	0.29%	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%



(a) ML - RMSE

(b) BC - RMSE

Fig. 5: RMSE performance for the (a) MovieLens and (b) BookCrossing data sets.



(a) ML - RMSE

(b) BC - RMSE

Fig. 6: Post hoc analysis for Friedman's Test of Accuracy Performance of different methods for the (a) MovieLens (ML) and (b) BookCrossing (BC) data sets.

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 F1 score. Table IV presents the results obtained by applying our method to the MovieLens and BookCrossing data sets. 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 F1 score for recommendation lists of size $k \in \{1, 3, 5, 10, 30, 50, 100\}$. Respectively, Fig. 7 illustrates the average performance over the same dimensions for lists of size $k \in \{1, 3, 5, 10, 20, \dots, 100\}$.

Table IV. : F1 Performance for the MovieLens and BookCrossing Data Sets.

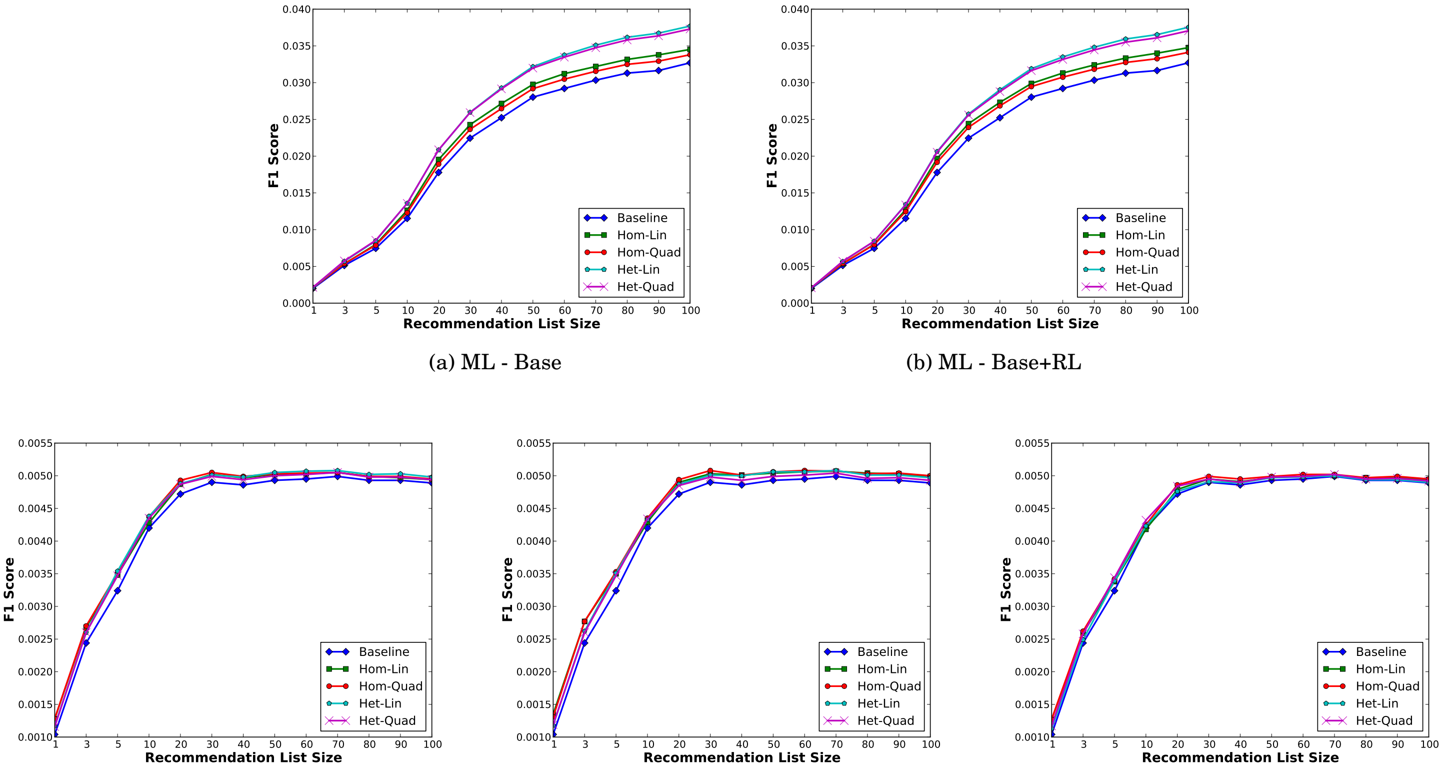
Data Set	**User Expectations**	**Experimental Setting**	**Recommendation List Size**						
			1	*3*	*5*	*10*	*30*	*50*	*100*
MovieLens	Base	Homogeneous Linear	5.00%	4.29%	7.10%	9.54%	8.15%	6.17%	5.57%
		Homogeneous Quadratic	4.00%	4.87%	5.63%	6.68%	5.35%	4.10%	3.36%
		Heterogeneous Linear	5.00%	10.92%	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%
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.

In particular, for the MovieLens data set and the case of the homogeneous users F1 score was improved by 6.14%, on average. In the case of heterogeneous customers performance was increased by 13.90%. For the BookCrossing data set, in the case of homogeneous users, F1 score was on average enhanced by 4.85% and, for heterogeneous users, by 3.16%.[11] 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 have tested the null hypothesis that the performance of each of the five lines of the graphs in Fig. 7 is the same using the Friedman test. Based on the results

[11] In Tables XIV - XVII of the 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

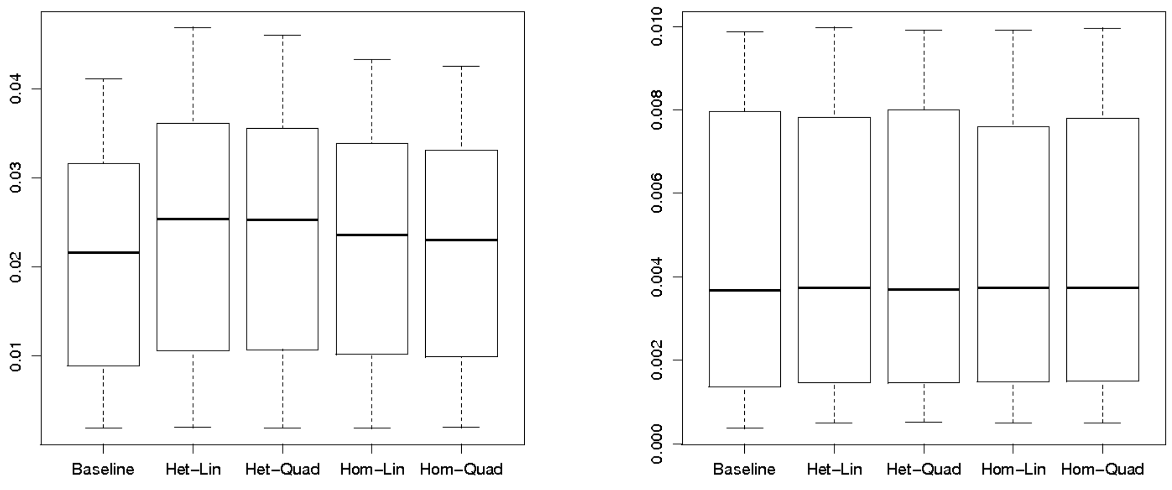
(e) BC - Base+AR

Fig. 7: F1 performance of different experimental settings for the (a), (b) MovieLens (ML) and (c), (d), (e) BookCrossing (BC) data sets.

we reject the null hypothesis with $p < 0.0001$. Performing post hoc analysis on Friedman's Test results for the ML data set, the differences between the *Baseline* and each one of the experimental settings are statistically significant for the F1 score. For the BC data set, the differences between the *Baseline* and each one of the experimental settings are also statistically significant.[12] Even though the lines are very close to each other and the differences in performance in absolute values are not large (e.g. Fig. 7e), the results are statistically significant since the performance of the proposed method is ranked consistently higher than the baselines (lines do not cross). Fig. 8 presents the box-and-whisker diagrams displaying the aforementioned differences among the various methods.

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.

[12] In the experiments conducted using the MovieLens data set, the difference between *Homogeneous Quadratic* and *Baseline* is statically significant with $p < 0.01$.



(a) MovieLens data set

(b) BookCrossing data set

Fig. 8: Post hoc analysis for Friedman's Test of F1 Performance of different methods for the (a) MovieLens and (b) BookCrossing data sets.

5.4 Comparison of Catalog Coverage and Aggregate Recommendation Diversity

In this section we investigate the effect of the proposed method for unexpected recommendations on coverage and aggregate diversity, two important metrics for RSs [Ge et al. 2010], [Adomavicius and Kwon 2012], [Shani and Gunawardana 2011].[13] 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: $|\bigcup_{u\in U}\mathrm{RS}_u|/\,|I|$) 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 Appendix present the results obtained by applying our method to the MovieLens and BookCrossing data sets. 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 summarizes these results over the different subsets. Fig. 9 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 and Fig. 9 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, if the recommender system can encourage users to rent "long-tail" movies because they are less costly to license and acquire from distributors than new-release or highly popular movies of big studios

[13] 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.

Table V. : Coverage Performance for the MovieLens and BookCrossing Data Sets.

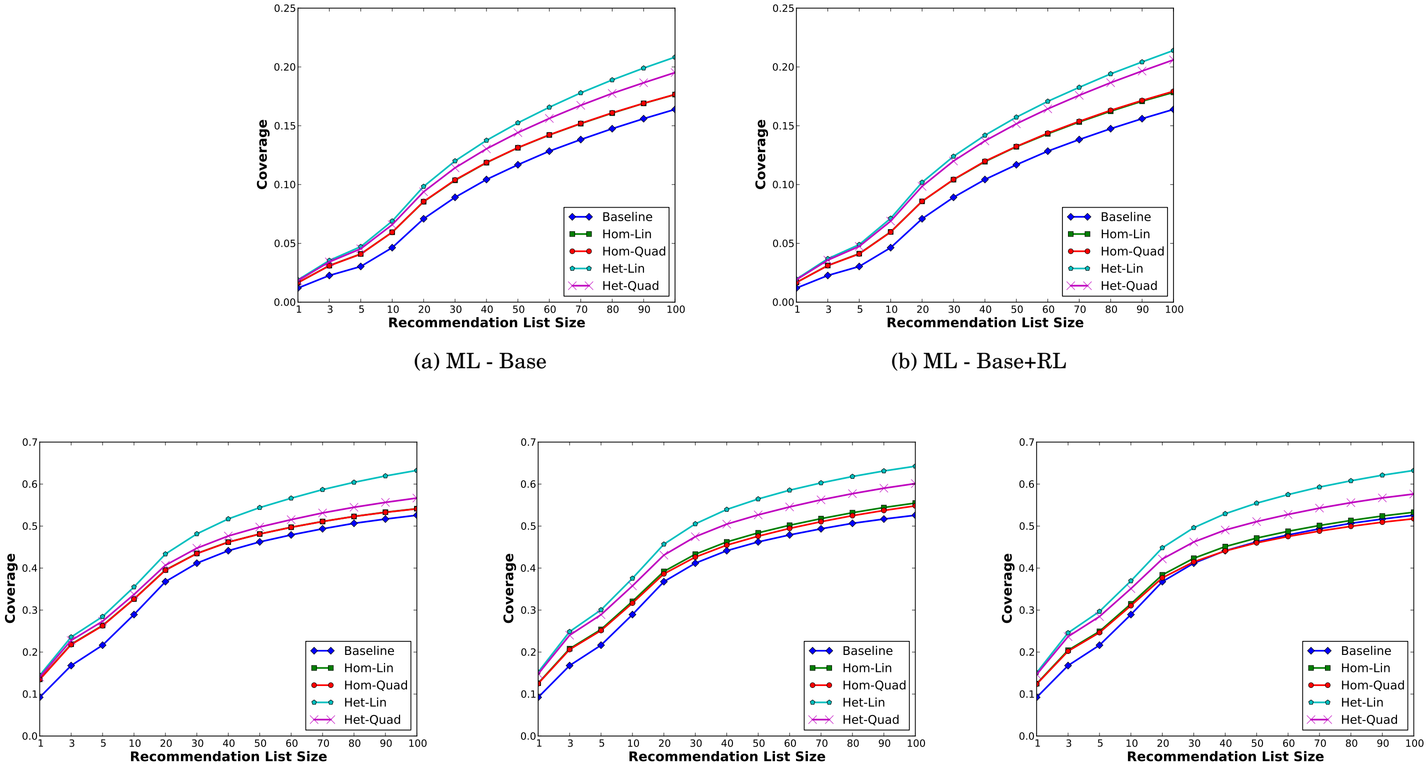
Data Set	**User Expectations**	**Experimental Setting**	**Recommendation List Size**						
			1	*3*	*5*	*10*	*30*	*50*	*100*
MovieLens	Base	Homogeneous Linear	38.58%	37.05%	35.15%	28.35%	16.27%	12.38%	7.70%
		Homogeneous Quadratic	38.41%	36.48%	34.65%	28.32%	16.62%	12.47%	7.77%
		Heterogeneous Linear	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.

[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 MovieLens data set 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 BookCrossing data set, 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. Besides, 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 MovieLens and BookCrossing data sets, 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 Fig. 9 is the same. Performing post hoc analysis on Friedman's Test results, for both the data sets the difference between the *Baseline* and each of the remaining experimental settings is statistically significant ($p < 0.001$). Fig. 10 presents the box-and-whisker diagrams displaying the aforementioned differences among the different methods.

The derived recommendation lists can also be evaluated for the inequality across items using the Gini coefficient [Gini 1909], the Hoover (Robin Hood) index [Hoover 1985], or the Lorenz curve [Lorenz 1905]. In particular, Fig. 11 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 Fig. 11, 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



(a) ML - Base

(b) ML - Base+RL

(c) BC - Base

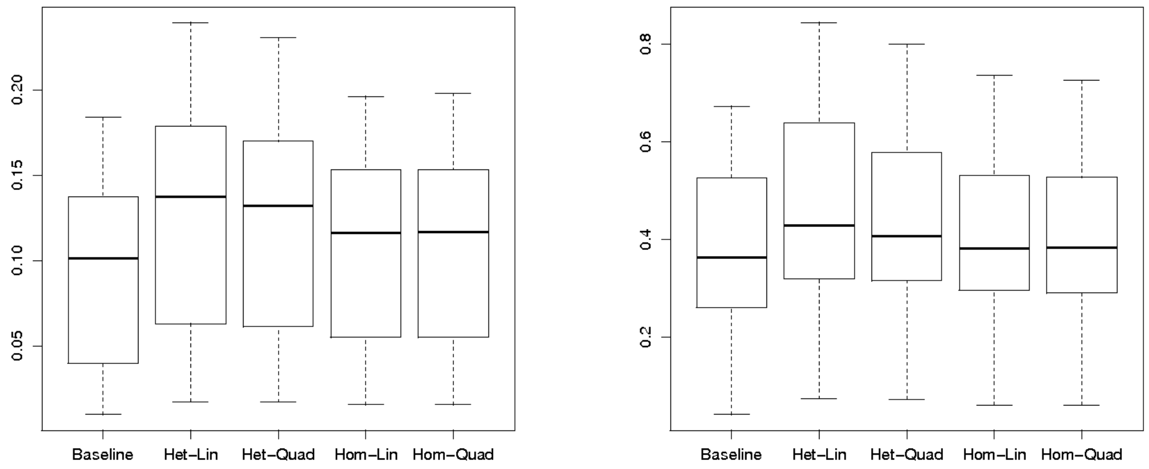
(d) BC - Base+RI

(e) BC - Base+AR

Fig. 9: Coverage performance of different experimental settings for the (a), (b) MovieLens (ML) and (c), (d), (e) BookCrossing (BC) data sets.

users are often capable of discovering by themselves. Hence, they are beneficial for both users and some organizations [Brynjolfsson et al. 2003; Brynjolfsson et al. 2011], [Goldstein and Goldstein 2006]. Finally, the difference in increase in performance between Figs. 11a and 11b, 0.98% and 7.17% respectively in terms of the Hoover index, could be attributed to both idiosyncrasies of the two data sets and the differences in definitions and cardinalities of the sets of expected recommendations discussed in Section 4.2.3.

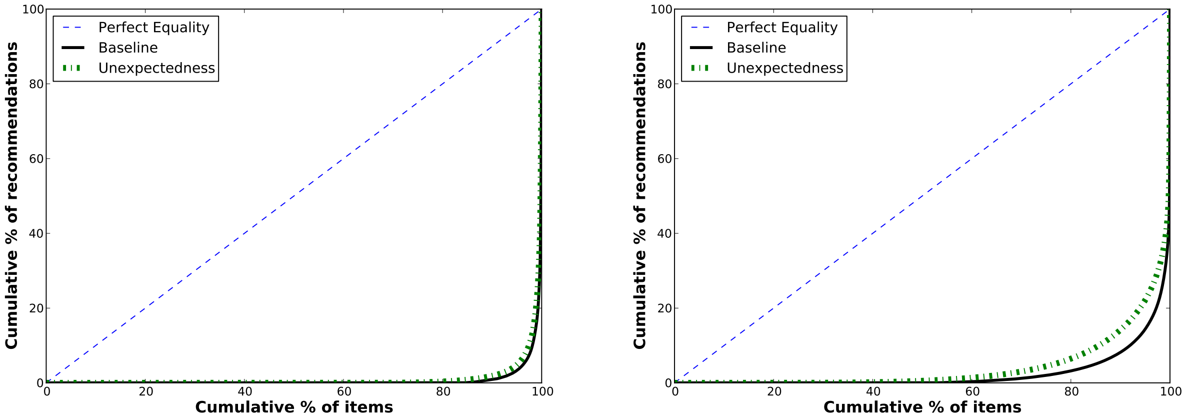
In summary, we demonstrated in this section that the proposed method for unexpected recommendations outperforms the standard baseline methods in terms of aggregate recommendation diversity and the classical catalog coverage measure.



(a) MovieLens data set

(b) BookCrossing data set

Fig. 10: Post hoc analysis for Friedman's Test of Coverage Performance of different methods for the (a) MovieLens and (b) BookCrossing data sets.



(a) ML - Base+RL

(b) BC - Base+RI

Fig. 11: Lorenz curves for recommendation lists of size k = 5 for the (a) MovieLens (ML) and (b) BookCrossing (BC) data sets.

6. DISCUSSION AND CONCLUSIONS

In this paper, we proposed and studied a concept of unexpected recommendations as recommending to a user those items that depart from what the specific user expects from the recommender system. After formally defining and formulating theoretically this concept, we operationalized the notion of *unexpectedness* and presented a method for providing unexpected recommendations of high quality based on users' utility that takes into account their interests. We also compared the generated unexpected recommendations with standard 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. Besides, we showed that the proposed method for unexpected recommendations also improves performance based on other important metrics, such as catalog coverage and aggregate diversity.

In particular, using different "real-world" data sets, 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, significantly outperformed in terms of unexpectedness the standard baseline algorithms, including item-based and user-based k-Nearest Neighbors, Slope One [Lemire and Maclachlan 2007], and Matrix Factorization [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 Collaborative Filtering 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 the classical accuracy-based measures, such as RMSE and F1 score.

One of the main premises of the proposed method is that users' expectations should be explicitly considered in order to provide the 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 the 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 data sets (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 on the small differences between the proposed metric of unexpectedness (Eq. 12) and the adapted metric of serendipity (Eq. 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. Further, 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 to the practitioners and researchers to adapt the proposed method to the recommendation settings of their own applications, by experimenting with different utility functions, estimation procedures, and sets of expectations, exploiting the domain knowledge they possess.

As a part of the future work, we would like to conduct live experiments with real users for evaluating unexpected recommendations. Moreover, we will further explore the notion of "monotonicity" introduced in Section 5.1 with the goal of formally and empirically demonstrating this effect. Further, we assumed in the experiments reported in the paper that a recommendation can be either expected or unexpected. We plan to relax this assumption in our future experiments.

APPENDIX

A. UNEXPECTEDNESS

Table VI. : Average Unexpectedness Performance for the MovieLens data set.

Data	**User**	**Experimental**	**Recommendation List Size**						
Subset	**Expectations**	**Setting**	*1*	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	1.59%	3.79%	4.43%	2.33%	1.67%	1.53%	1.03%
		Homogeneous Quadratic	1.59%	3.34%	3.99%	2.32%	1.63%	1.50%	1.08%
		Heterogeneous Linear	2.03%	2.45%	2.89%	1.85%	1.20%	1.07%	0.72%
		Heterogeneous Quadratic	1.77%	2.06%	2.39%	1.52%	1.01%	0.93%	0.64%
	Base+RL	Homogeneous Linear	17.41%	19.70%	14.40%	10.15%	9.54%	9.58%	7.72%
		Homogeneous Quadratic	14.07%	17.07%	13.40%	10.12%	9.26%	9.28%	7.70%
		Heterogeneous Linear	14.40%	13.50%	10.89%	7.94%	6.85%	6.85%	5.69%
		Heterogeneous Quadratic	11.82%	11.31%	9.09%	6.86%	5.97%	6.09%	5.15%
b	Base	Homogeneous Linear	2.21%	3.35%	3.44%	2.27%	1.81%	1.49%	1.13%
		Homogeneous Quadratic	2.03%	3.32%	3.27%	2.49%	1.92%	1.67%	1.24%
		Heterogeneous Linear	1.50%	2.04%	2.03%	1.87%	1.55%	1.36%	1.02%
		Heterogeneous Quadratic	1.46%	1.93%	2.02%	1.83%	1.54%	1.34%	1.03%
	Base+RL	Homogeneous Linear	24.63%	16.90%	17.90%	15.43%	11.65%	10.45%	8.26%
		Homogeneous Quadratic	22.05%	18.35%	19.34%	17.20%	13.67%	12.62%	10.48%
		Heterogeneous Linear	18.07%	16.28%	16.07%	14.87%	12.00%	11.30%	9.50%
		Heterogeneous Quadratic	17.32%	15.94%	15.84%	14.70%	12.05%	11.35%	9.68%

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

Table VII. : Average Serendipity Performance for the MovieLens data set.

Data	**User**	**Experimental**	**Recommendation List Size**						
Subset	**Expectations**	**Setting**	*1*	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	1.63%	3.84%	4.48%	2.36%	1.69%	1.54%	1.05%
		Homogeneous Quadratic	1.63%	3.39%	4.03%	2.34%	1.65%	1.52%	1.09%
		Heterogeneous Linear	2.08%	2.49%	2.94%	1.88%	1.23%	1.09%	0.75%
		Heterogeneous Quadratic	1.81%	2.10%	2.43%	1.55%	1.02%	0.94%	0.66%
	Base+RL	Homogeneous Linear	17.50%	19.80%	14.48%	10.19%	9.56%	9.61%	7.74%
		Homogeneous Quadratic	14.15%	17.15%	13.46%	10.16%	9.28%	9.31%	7.71%
		Heterogeneous Linear	14.48%	13.58%	10.96%	7.98%	6.88%	6.88%	5.71%
		Heterogeneous Quadratic	11.90%	11.38%	9.15%	6.90%	6.00%	6.11%	5.18%
b	Base	Homogeneous Linear	2.21%	3.35%	3.44%	2.28%	1.81%	1.49%	1.13%
		Homogeneous Quadratic	2.03%	3.32%	3.27%	2.49%	1.92%	1.67%	1.24%
		Heterogeneous Linear	1.50%	2.04%	2.03%	1.87%	1.55%	1.36%	1.02%
		Heterogeneous Quadratic	1.46%	1.93%	2.02%	1.83%	1.54%	1.34%	1.03%
	Base+RL	Homogeneous Linear	24.63%	16.90%	17.90%	15.42%	11.65%	10.45%	8.26%
		Homogeneous Quadratic	22.05%	18.34%	19.34%	17.20%	13.67%	12.62%	10.48%
		Heterogeneous Linear	18.07%	16.28%	16.07%	14.87%	12.00%	11.30%	9.50%
		Heterogeneous Quadratic	17.32%	15.94%	15.84%	14.69%	12.05%	11.35%	9.67%

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

Table VIII. : Average Unexpectedness Performance for the BookCrossing data set.

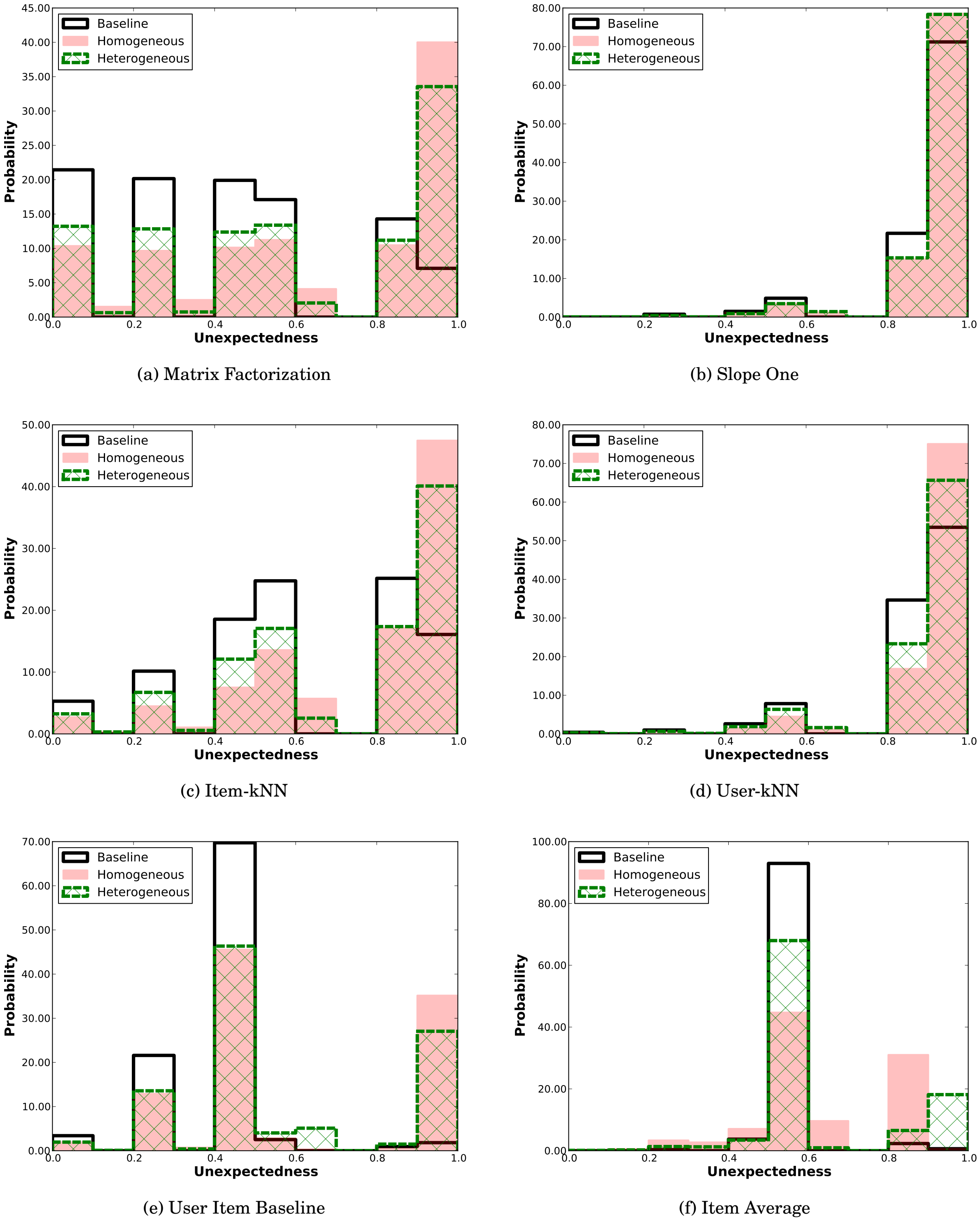
Data	User	Experimental	Recommendation List Size						
Subset	**Expectations**	**Setting**	*1*	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	0.29%	0.32%	0.29%	0.26%	0.23%	0.21%	0.19%
		Homogeneous Quadratic	0.28%	0.31%	0.27%	0.24%	0.21%	0.19%	0.17%
		Heterogeneous Linear	0.16%	0.15%	0.13%	0.12%	0.11%	0.10%	0.09%
		Heterogeneous Quadratic	0.15%	0.13%	0.12%	0.10%	0.09%	0.09%	0.08%
	Base+RI	Homogeneous Linear	153.68%	102.31%	102.10%	92.09%	70.32%	62.04%	53.33%
		Homogeneous Quadratic	151.56%	100.90%	100.60%	91.14%	69.49%	61.40%	52.99%
		Heterogeneous Linear	60.17%	41.43%	46.85%	35.32%	28.13%	27.20%	23.75%
		Heterogeneous Quadratic	54.19%	37.28%	43.18%	31.63%	25.46%	25.00%	21.94%
	Base+AR	Homogeneous Linear	143.64%	96.11%	95.73%	86.16%	65.05%	57.05%	48.82%
		Homogeneous Quadratic	141.74%	94.84%	94.35%	85.29%	64.33%	56.49%	48.57%
		Heterogeneous Linear	56.87%	39.44%	44.47%	33.19%	26.04%	25.15%	21.87%
		Heterogeneous Quadratic	51.38%	35.63%	41.12%	29.79%	23.60%	23.16%	20.25%
b	Base	Homogeneous Linear	0.89%	0.72%	0.66%	0.72%	0.64%	0.59%	0.52%
		Homogeneous Quadratic	0.49%	0.49%	0.44%	0.42%	0.35%	0.31%	0.26%
		Heterogeneous Linear	0.36%	0.36%	0.32%	0.37%	0.36%	0.36%	0.35%
		Heterogeneous Quadratic	0.33%	0.34%	0.30%	0.34%	0.33%	0.34%	0.33%
	Base+RI	Homogeneous Linear	212.54%	226.48%	197.24%	176.14%	156.18%	148.83%	140.99%
		Homogeneous Quadratic	215.58%	228.31%	200.36%	179.59%	157.51%	149.69%	138.85%
		Heterogeneous Linear	132.74%	148.06%	129.70%	117.33%	103.92%	101.22%	95.69%
		Heterogeneous Quadratic	124.67%	140.24%	122.86%	111.03%	98.27%	95.96%	90.73%
	Base+AR	Homogeneous Linear	163.59%	178.81%	156.47%	141.58%	123.70%	118.83%	108.17%
		Homogeneous Quadratic	165.12%	180.81%	160.07%	142.96%	123.16%	116.41%	104.34%
		Heterogeneous Linear	102.61%	117.47%	103.54%	93.18%	81.13%	78.52%	71.74%
		Heterogeneous Quadratic	96.09%	110.84%	97.55%	87.74%	76.25%	73.97%	67.65%
c	Base	Homogeneous Linear	1.51%	1.66%	1.60%	1.55%	1.52%	1.54%	1.51%
		Homogeneous Quadratic	1.09%	1.16%	1.14%	1.03%	1.00%	1.01%	0.98%
		Heterogeneous Linear	0.78%	0.88%	0.86%	0.84%	0.87%	0.89%	0.91%
		Heterogeneous Quadratic	0.69%	0.79%	0.79%	0.76%	0.80%	0.82%	0.84%
	Base+RI	Homogeneous Linear	195.48%	181.47%	179.07%	162.41%	161.52%	149.87%	118.69%
		Homogeneous Quadratic	202.85%	193.20%	192.25%	176.43%	176.21%	164.73%	132.44%
		Heterogeneous Linear	98.06%	91.39%	90.86%	84.68%	84.89%	79.69%	64.20%
		Heterogeneous Quadratic	91.07%	84.92%	84.46%	78.92%	79.37%	74.69%	60.44%
	Base+AR	Homogeneous Linear	170.06%	155.50%	152.71%	137.46%	134.23%	123.24%	95.59%
		Homogeneous Quadratic	175.59%	164.67%	162.66%	148.29%	145.07%	133.58%	104.97%
		Heterogeneous Linear	86.66%	78.51%	77.27%	71.18%	69.53%	64.21%	50.45%
		Heterogeneous Quadratic	80.53%	72.83%	71.74%	66.21%	64.85%	60.00%	47.33%

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

Table IX. : Average Serendipity Performance for the BookCrossing data set.

Data	**User**	**Experimental**	**Recommendation List Size**						
Subset	**Expectations**	**Setting**	*1*	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	0.26%	0.40%	0.40%	0.31%	0.23%	0.21%	0.22%
		Homogeneous Quadratic	0.24%	0.39%	0.39%	0.30%	0.21%	0.19%	0.20%
		Heterogeneous Linear	0.15%	0.25%	0.27%	0.20%	0.12%	0.10%	0.12%
		Heterogeneous Quadratic	0.13%	0.22%	0.25%	0.18%	0.11%	0.08%	0.10%
	Base+RI	Homogeneous Linear	153.58%	102.75%	102.71%	92.42%	70.42%	62.11%	53.48%
		Homogeneous Quadratic	151.46%	101.32%	101.19%	91.46%	69.59%	61.47%	53.09%
		Heterogeneous Linear	60.17%	41.82%	47.37%	35.62%	28.22%	27.25%	23.85%
		Heterogeneous Quadratic	54.19%	37.65%	43.69%	31.93%	25.54%	25.04%	22.03%
	Base+AR	Homogeneous Linear	143.57%	96.49%	96.26%	86.42%	65.10%	57.08%	48.92%
		Homogeneous Quadratic	141.65%	95.20%	94.87%	85.55%	64.37%	56.52%	48.61%
		Heterogeneous Linear	56.87%	39.80%	44.97%	33.48%	26.12%	25.19%	21.96%
		Heterogeneous Quadratic	51.39%	35.98%	41.61%	30.07%	23.68%	23.20%	20.34%
b	Base	Homogeneous Linear	-0.34%	-0.10%	0.17%	0.48%	0.57%	0.55%	0.49%
		Homogeneous Quadratic	-0.74%	-0.33%	-0.05%	0.17%	0.27%	0.26%	0.24%
		Heterogeneous Linear	-0.86%	-0.46%	-0.16%	0.13%	0.28%	0.32%	0.33%
		Heterogeneous Quadratic	-0.90%	-0.48%	-0.19%	0.10%	0.26%	0.30%	0.31%
	Base+RI	Homogeneous Linear	206.87%	222.49%	195.02%	175.12%	155.93%	148.70%	140.96%
		Homogeneous Quadratic	209.87%	224.30%	198.14%	178.57%	157.26%	149.56%	138.82%
		Heterogeneous Linear	127.07%	144.07%	127.49%	116.31%	103.66%	101.07%	95.64%
		Heterogeneous Quadratic	119.01%	136.25%	120.64%	110.01%	98.00%	95.81%	90.68%
	Base+AR	Homogeneous Linear	158.93%	175.47%	154.60%	140.71%	123.47%	118.72%	108.14%
		Homogeneous Quadratic	160.42%	177.45%	158.20%	142.10%	122.94%	116.29%	104.31%
		Heterogeneous Linear	97.97%	114.15%	101.68%	92.32%	80.90%	78.40%	71.69%
		Heterogeneous Quadratic	91.45%	107.51%	95.69%	86.88%	76.02%	73.85%	67.60%
c	Base	Homogeneous Linear	1.51%	1.66%	1.60%	1.55%	1.52%	1.54%	1.51%
		Homogeneous Quadratic	1.09%	1.16%	1.14%	1.03%	1.00%	1.01%	0.98%
		Heterogeneous Linear	0.78%	0.88%	0.86%	0.84%	0.87%	0.89%	0.91%
		Heterogeneous Quadratic	0.69%	0.79%	0.79%	0.76%	0.80%	0.82%	0.84%
	Base+RI	Homogeneous Linear	195.48%	181.54%	179.10%	162.44%	161.54%	149.89%	118.70%
		Homogeneous Quadratic	202.85%	193.27%	192.28%	176.46%	176.22%	164.74%	132.45%
		Heterogeneous Linear	98.06%	91.43%	90.88%	84.70%	84.89%	79.70%	64.20%
		Heterogeneous Quadratic	91.07%	84.96%	84.48%	78.94%	79.38%	74.70%	60.44%
	Base+AR	Homogeneous Linear	170.06%	155.55%	152.73%	137.48%	134.24%	123.25%	95.59%
		Homogeneous Quadratic	175.59%	164.71%	162.68%	148.32%	145.08%	133.59%	104.97%
		Heterogeneous Linear	86.66%	78.53%	77.28%	71.19%	69.53%	64.21%	50.45%
		Heterogeneous Quadratic	80.53%	72.85%	71.75%	66.23%	64.86%	60.00%	47.33%

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



(a) Matrix Factorization

(b) Slope One

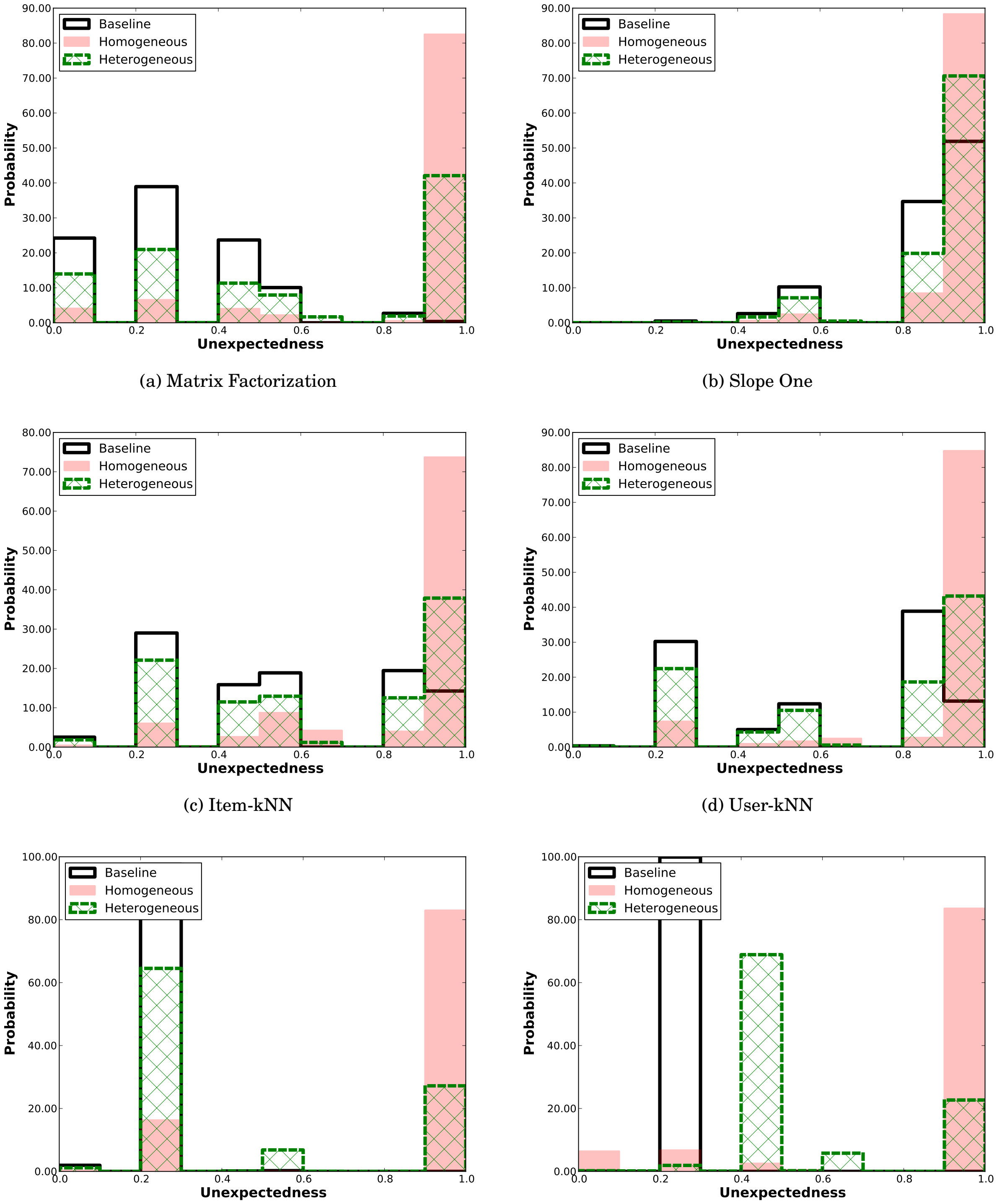
(c) Item-kNN

(d) User-kNN

(e) User Item Baseline

(f) Item Average

Fig. 12: Distribution of Unexpectedness for recommendation lists of size $k = 5$ and different baseline algorithms for the MovieLens data sets using the (Base+RL) set of user expectations.



(a) Matrix Factorization

(b) Slope One

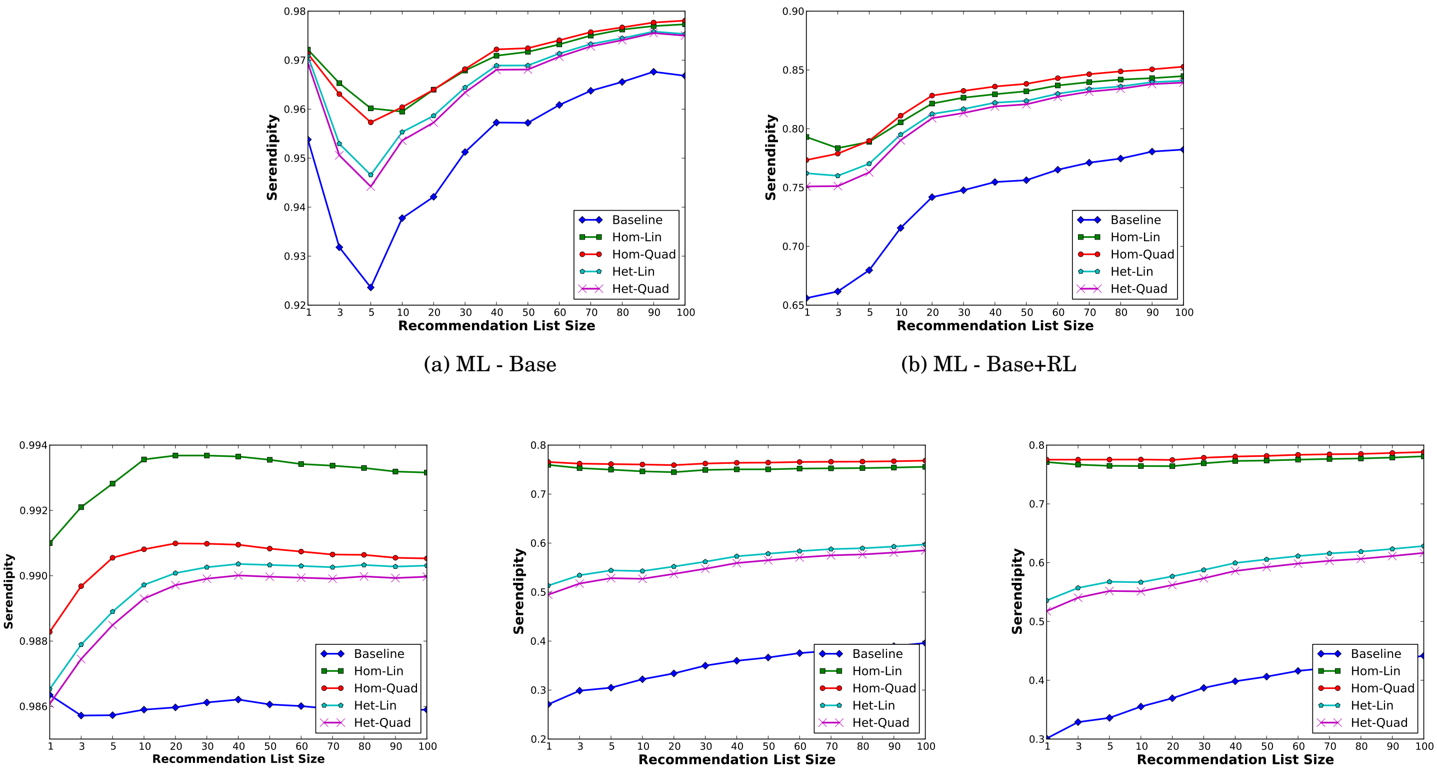
(c) Item-kNN

(d) User-kNN

(e) User Item Baseline

(f) Item Average

Fig. 13: Distribution of Unexpectedness for recommendation lists of size $k = 5$ and different baseline algorithms for the BookCrossing data sets using the (Base+RI) set of user expectations.



(a) ML - Base

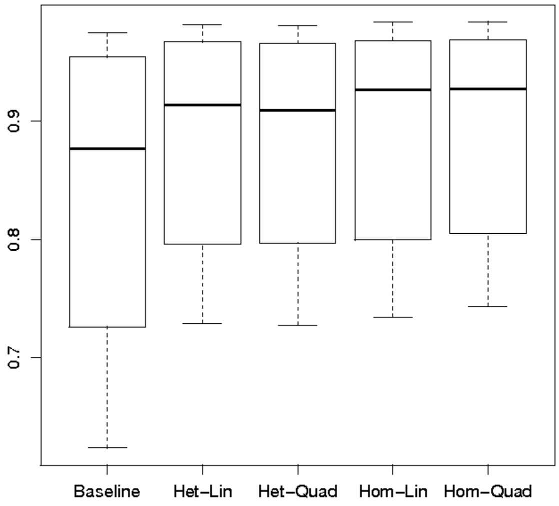
(b) ML - Base+RL

(c) BC - Base

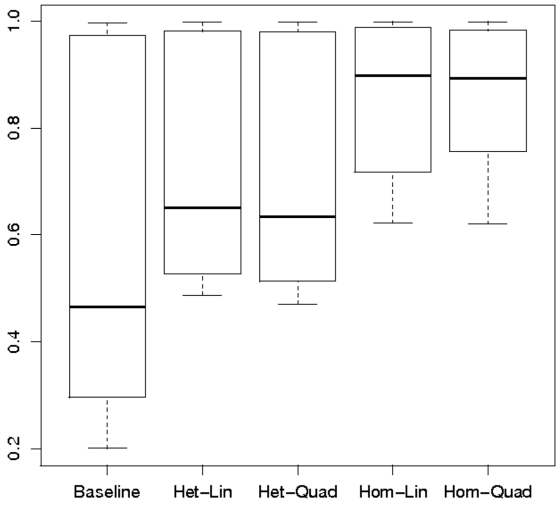
(d) BC - Base+RI

(e) BC - Base+AR

Fig. 14: Serendipity performance of different experimental settings for the (a), (b) MovieLens (ML) and (c), (d), (e) BookCrossing (BC) data sets.



(a) MovieLens data set



(b) BookCrossing data set

Fig. 15: Post hoc analysis for Friedman's Test of Serendipity Performance of different methods for the (a) MovieLens and (b) BookCrossing data sets.

A.1 Qualitative Comparison of Unexpectedness

We present here some recommendation examples, additional to those presented in Section 5.1.1, in order to further evaluate the proposed approach.

Using the MovieLens data set and the (Base+RL) set of expected recommendations described in Section 4.2.3, the baseline methods recommend to a user, who has highly rated a large number of very popular Action, Crime, Drama, Thriller, and War films, the movies “The Shawshank Redemption”, “The Usual Suspects”, and “The Godfather” (user id = 13221 with Item-based kNN). However, the specific user has already highly rated many closely related movies (i.e. common cast, user tags, etc.) such as “The Bucket List”, “American Beauty”, “The Life of David Gale”, “The Silence of the Lambs”, and “The Matrix”. Hence, the aforementioned popular recommendations are highly expected for the specific user. On the other hand, the proposed algorithm recommends the following unexpected movies: “Shichinin no samurai”, “Das Leben der Anderen”, and “One Day in September”. These movies are of high quality, unexpected, not irrelevant to the user, and they fairly match the user’s interests as indicated by rating highly movies such as “Kagemusha”, “Nausicaä of the Valley of the Wind”, “Lord of War”, “Charlie Wilson’s War”, “Das Boot”, and others. Interestingly enough, these recommendations are based on movies that they have been filmed by the same director and they belong to different genres (i.e. “Kagemusha” and “Shichinin no samurai”) or they involve elements in their plot, such as history, war and police, that can be also found in other films that the specific user likes.

Using the BookCrossing data set and the (Base) set of expectations described in Section 4.2.3, the baseline method [Koren 2010] recommends to a user the following highly expected books: “Harry Potter and the Chamber of Secrets”, “To Kill a Mockingbird”, and “Lord of the Rings: The Two Towers” (user id = 235842). However, the specific user is already aware of and familiar with these items (i.e. implicit rating). Hence, the aforementioned popular recommendations are totally expected for the specific user. On the other hand, for the same user, the proposed method generated the following recommendations: “84, Charing Cross Road”, “Tell No One”, and “Night”. These less popular recommendations are not only of great quality, but also unexpected for the specific user while they still provide a fair match to her/his interests. In particular, these Biography, History, Mystery, Literature, and Fiction books, even though being unexpected for the user, they are not irrelevant and they fairly match the user’s profile since she/he has already highly rated books such as “Embers”, “Plain Truth”, “A Time to Kill”, and “Bringing Elizabeth Home” which deal with hope, faith, survival, interpersonal relations, cultural differences, racism, crimes or mystery.

Respectively, using the (Base+AR) set of expectations, the baseline methods recommend to a user, who has highly rated Literature, Fiction, and Mystery books, the items “The Five People You Meet in Heaven”, “1st to Die: A Novel”, and “The Da Vinci Code” (user id = 2099 with Item-based kNN). However, based on the mechanisms described in detail in Section 4.2.3, these recommendations are expected for the specific user since she/he has already rated the books “The Notebook”, “The Red Tent”, and “The Dive From Clausen’s Pier”. Nevertheless, the proposed algorithm recommends to the user the following unexpected books: “My Sister’s Keeper: A Novel”, “The Devil in the White City”, and “The Curious Incident of the Dog in the Night-Time”. All of these books are both of great quality and significantly depart from the expectations of the user. Also, they are not irrelevant and they fairly match the user’s interests since all these recommendations deal with interpersonal relations, family, religion, values, or mystery; the user has already highly rated books such as “The Swallows of Kabul : A Novel” “Road Less Traveled: A New Psychology of Love, Traditional Values, and Spritual Growth” “A Lesson Before Dying”, “The Final Judgment”, and “Pleading Guilty”.

APPENDIX

B. RATING PREDICTION

B.1 RMSE

Table X. : RMSE Performance for the MovieLens data set.

Rating Prediction Algorithm	Subset	Expectations	Baseline	Homogeneous		Heterogeneous	
				Linear	Quadratic	Linear	Quadratic
MatrixFactorization	a	Base	0.7934	0.27%	0.28%	0.22%	0.27%
		Base+RL	0.7934	0.27%	0.28%	0.23%	0.27%
	b	Base	0.7849	-0.04%	-0.03%	-0.08%	-0.03%
		Base+RL	0.7849	-0.04%	-0.02%	-0.08%	-0.03%
SlopeOne	a	Base	0.8286	0.57%	0.57%	0.75%	0.74%
		Base+RL	0.8286	0.57%	0.57%	0.74%	0.73%
	b	Base	0.8198	0.01%	0.01%	0.11%	0.12%
		Base+RL	0.8198	0.01%	0.01%	0.11%	0.11%
ItemKNN	a	Base	0.8103	-0.01%	0.00%	0.01%	0.02%
		Base+RL	0.8103	-0.01%	0.00%	0.02%	0.02%
	b	Base	0.8083	-0.02%	-0.01%	-0.01%	0.01%
		Base+RL	0.8083	-0.01%	-0.01%	0.00%	0.01%
UserKNN	a	Base	0.8174	0.01%	0.01%	0.03%	0.05%
		Base+RL	0.8174	0.01%	0.01%	0.04%	0.04%
	b	Base	0.8146	0.01%	0.01%	0.02%	0.04%
		Base+RL	0.8146	0.01%	0.01%	0.02%	0.03%
UserItemBaseline	a	Base	0.8265	0.01%	0.01%	0.06%	0.07%
		Base+RL	0.8265	0.01%	0.01%	0.08%	0.07%
	b	Base	0.8246	0.00%	0.00%	0.02%	0.03%
		Base+RL	0.8246	0.00%	0.00%	0.03%	0.04%
ItemAverage	a	Base	0.8952	0.01%	0.01%	1.38%	1.66%
		Base+RL	0.8952	0.02%	0.01%	1.40%	1.72%
	b	Base	0.8913	0.01%	0.00%	1.13%	1.39%
		Base+RL	0.8913	0.01%	0.01%	1.19%	1.42%

Table XI. : RMSE Performance for the BookCrossing data set.

Rating Prediction Algorithm	Subset	Expectations	Baseline	Homogeneous		Heterogeneous	
				Linear	Quadratic	Linear	Quadratic
MatrixFactorization	a	Base	1.8134	0.20%	0.28%	-0.62%	-0.14%
		Base+RI	1.8134	-0.19%	-0.52%	-0.78%	-0.03%
		Base+AS	1.8134	-0.34%	-0.53%	-0.84%	-0.03%
	b	Base	1.7453	0.66%	0.72%	0.06%	0.29%
		Base+RI	1.7453	0.58%	0.65%	0.06%	0.36%
		Base+AS	1.7453	0.55%	0.64%	0.04%	0.34%
	c	Base	1.8059	-0.01%	0.06%	-0.49%	-0.09%
		Base+RI	1.8059	-0.25%	-0.55%	-0.56%	-0.29%
		Base+AS	1.8059	-0.19%	-0.54%	-0.56%	-0.33%
SlopeOne	a	Base	2.0066	4.39%	4.47%	3.78%	4.07%
		Base+RI	2.0066	4.12%	3.65%	3.70%	4.11%
		Base+AS	2.0066	4.03%	3.66%	3.69%	4.11%
	b	Base	1.8371	3.93%	3.99%	3.01%	3.59%
		Base+RI	1.8371	3.75%	3.91%	2.97%	3.60%
		Base+AS	1.8371	3.76%	3.91%	2.99%	3.60%
	c	Base	1.7317	1.99%	2.10%	0.96%	1.69%
		Base+RI	1.7317	1.58%	1.47%	0.28%	0.67%
		Base+AS	1.7317	1.86%	1.55%	0.44%	1.03%
ItemKNN	a	Base	1.6512	1.31%	1.30%	-2.12%	-0.89%
		Base+RI	1.6512	1.17%	0.32%	-1.99%	-0.67%
		Base+AS	1.6512	1.31%	0.32%	-2.16%	-0.70%
	b	Base	1.5796	1.04%	1.03%	-1.18%	-0.34%
		Base+RI	1.5796	1.02%	0.98%	-1.25%	-0.39%
		Base+AS	1.5796	1.02%	0.98%	-1.38%	-0.45%
	c	Base	1.6436	2.04%	2.03%	-0.34%	0.54%
		Base+RI	1.6436	2.11%	1.76%	-1.09%	-0.72%
		Base+AS	1.6436	2.12%	1.77%	-1.02%	-0.48%
UserKNN	a	Base	1.7465	1.34%	0.79%	-0.93%	-0.09%
		Base+RI	1.7465	1.47%	0.52%	-0.97%	0.01%
		Base+AS	1.7465	1.50%	0.52%	-1.03%	0.01%
	b	Base	1.6924	1.03%	1.00%	-0.50%	0.05%
		Base+RI	1.6924	1.04%	0.98%	-0.40%	0.25%
		Base+AS	1.6924	1.05%	0.99%	-0.47%	0.20%
	c	Base	1.7450	1.86%	1.79%	0.20%	0.77%
		Base+RI	1.7450	1.81%	1.48%	-0.62%	-0.31%
		Base+AS	1.7450	1.84%	1.51%	-0.30%	0.08%
UserItemBaseline	a	Base	1.6350	3.01%	2.63%	0.37%	1.46%
		Base+RI	1.6350	1.97%	1.66%	0.20%	1.45%
		Base+AS	1.6350	1.99%	1.66%	0.02%	1.43%
	b	Base	1.5322	1.98%	1.96%	0.17%	0.67%
		Base+RI	1.5322	1.92%	1.91%	0.18%	0.87%
		Base+AS	1.5322	1.93%	1.91%	0.02%	0.75%
	c	Base	1.5667	2.44%	2.43%	0.10%	0.84%
		Base+RI	1.5667	1.91%	1.73%	-0.80%	-0.29%
		Base+AS	1.5667	2.01%	1.76%	-0.47%	-0.05%
ItemAverage	a	Base	1.8714	-0.13%	-0.58%	-0.55%	0.39%
		Base+RI	1.8714	-0.39%	-0.94%	-0.58%	0.49%
		Base+AS	1.8714	-0.25%	-0.92%	-0.67%	0.48%
	b	Base	1.7146	0.01%	-0.01%	0.48%	1.00%
		Base+RI	1.7146	-0.03%	-0.05%	0.59%	1.20%
		Base+AS	1.7146	-0.02%	-0.05%	0.45%	1.10%
	c	Base	1.6984	0.33%	0.29%	-0.45%	0.10%
		Base+RI	1.6984	0.29%	0.01%	-0.85%	-0.01%
		Base+AS	1.6984	0.29%	-0.25%	-0.82%	-0.07%

B.2 MAE

Table XII. : MAE Performance for the MovieLens data set.

Rating Prediction Algorithm	Subset	Expectations	Baseline	Homogeneous		Heterogeneous	
				Linear	Quadratic	Linear	Quadratic
MatrixFactorization	a	Base	0.6090	0.23%	0.25%	0.26%	0.28%
		Base+RL	0.6090	0.23%	0.25%	0.27%	0.28%
	b	Base	0.6034	0.06%	0.07%	0.08%	0.10%
		Base+RL	0.6034	0.06%	0.07%	0.09%	0.10%
SlopeOne	a	Base	0.6378	0.74%	0.74%	0.96%	0.97%
		Base+RL	0.6378	0.74%	0.74%	0.95%	0.95%
	b	Base	0.6314	0.15%	0.15%	0.29%	0.31%
		Base+RL	0.6314	0.16%	0.16%	0.29%	0.30%
ItemKNN	a	Base	0.6230	0.03%	0.04%	0.15%	0.15%
		Base+RL	0.6230	0.03%	0.04%	0.15%	0.15%
	b	Base	0.6212	0.03%	0.05%	0.13%	0.13%
		Base+RL	0.6212	0.03%	0.05%	0.14%	0.13%
UserKNN	a	Base	0.6285	-0.06%	-0.06%	0.09%	0.10%
		Base+RL	0.6285	-0.06%	-0.06%	0.10%	0.10%
	b	Base	0.6264	-0.02%	-0.02%	0.08%	0.09%
		Base+RL	0.6264	-0.02%	-0.02%	0.09%	0.09%
UserItemBaseline	a	Base	0.6376	0.07%	0.07%	0.14%	0.14%
		Base+RL	0.6376	0.07%	0.07%	0.15%	0.15%
	b	Base	0.6353	0.07%	0.07%	0.11%	0.11%
		Base+RL	0.6353	0.07%	0.07%	0.12%	0.12%
ItemAverage	a	Base	0.6905	-0.02%	-0.01%	2.15%	2.58%
		Base+RL	0.6905	-0.01%	0.00%	2.19%	2.68%
	b	Base	0.6874	-0.02%	-0.01%	1.79%	2.18%
		Base+RL	0.6874	-0.02%	-0.01%	1.88%	2.24%

Table XIII. : MAE Performance for the BookCrossing data set.

Rating Prediction Algorithm	Subset	Expectations	Baseline	Homogeneous		Heterogeneous	
				Linear	Quadratic	Linear	Quadratic
MatrixFactorization	a	Base	1.3724	0.91%	0.99%	0.24%	0.60%
		Base+RI	1.3724	0.73%	0.66%	0.12%	0.76%
		Base+AS	1.3724	0.67%	0.65%	0.12%	0.78%
	b	Base	1.2994	1.28%	1.35%	0.72%	1.02%
		Base+RI	1.2994	1.20%	1.28%	0.77%	1.18%
		Base+AS	1.2994	1.19%	1.27%	0.73%	1.17%
	c	Base	1.3455	1.16%	1.07%	0.49%	0.96%
		Base+RI	1.3455	1.00%	0.90%	0.55%	0.92%
		Base+AS	1.3455	1.05%	0.92%	0.60%	0.87%
SlopeOne	a	Base	1.4915	4.38%	4.48%	3.89%	4.23%
		Base+RI	1.4915	4.13%	3.97%	3.82%	4.27%
		Base+AS	1.4915	4.10%	3.98%	3.84%	4.28%
	b	Base	1.3499	4.20%	4.29%	3.33%	4.01%
		Base+RI	1.3499	4.01%	4.19%	3.26%	4.03%
		Base+AS	1.3499	4.02%	4.19%	3.26%	4.05%
	c	Base	1.2862	2.64%	2.72%	1.76%	2.68%
		Base+RI	1.2862	2.33%	2.37%	1.16%	2.01%
		Base+AS	1.2862	2.51%	2.47%	1.31%	2.20%
ItemKNN	a	Base	1.2353	1.32%	1.30%	-1.04%	0.04%
		Base+RI	1.2353	1.08%	0.66%	-1.03%	0.23%
		Base+AS	1.2353	1.17%	0.68%	-1.08%	0.23%
	b	Base	1.1466	0.22%	0.22%	-1.51%	-0.47%
		Base+RI	1.1466	0.20%	0.18%	-1.74%	-0.53%
		Base+AS	1.1466	0.20%	0.18%	-1.84%	-0.57%
	c	Base	1.1939	1.63%	1.61%	-0.46%	0.70%
		Base+RI	1.1939	1.62%	1.43%	-1.28%	-0.25%
		Base+AS	1.1939	1.63%	1.45%	-1.13%	-0.08%
UserKNN	a	Base	1.3319	1.28%	0.84%	0.21%	0.96%
		Base+RI	1.3319	1.25%	0.73%	0.27%	1.17%
		Base+AS	1.3319	1.29%	0.74%	0.27%	1.18%
	b	Base	1.2793	0.84%	0.82%	0.27%	0.97%
		Base+RI	1.2793	0.84%	0.80%	0.34%	1.16%
		Base+AS	1.2793	0.85%	0.81%	0.25%	1.13%
	c	Base	1.3199	2.53%	2.48%	1.59%	2.41%
		Base+RI	1.3199	2.54%	2.36%	0.82%	1.66%
		Base+AS	1.3199	2.56%	2.39%	1.02%	1.92%
UserItemBaseline	a	Base	1.2496	1.62%	1.51%	0.77%	1.68%
		Base+RI	1.2496	1.04%	0.99%	0.81%	1.90%
		Base+AS	1.2496	1.12%	1.02%	0.75%	1.91%
	b	Base	1.1791	2.19%	2.19%	1.78%	2.58%
		Base+RI	1.1791	2.09%	2.11%	1.78%	2.80%
		Base+AS	1.1791	2.11%	2.09%	1.70%	2.78%
	c	Base	1.2014	2.83%	2.74%	1.67%	2.60%
		Base+RI	1.2014	2.49%	2.35%	1.05%	2.09%
		Base+AS	1.2014	2.59%	2.45%	1.32%	2.27%
ItemAverage	a	Base	1.4650	-0.19%	-0.42%	1.19%	2.10%
		Base+RI	1.4650	-0.40%	-0.63%	1.26%	2.34%
		Base+AS	1.4650	-0.29%	-0.61%	1.18%	2.33%
	b	Base	1.3428	-0.01%	-0.01%	2.43%	3.34%
		Base+RI	1.3428	-0.06%	-0.07%	2.58%	3.63%
		Base+AS	1.3428	-0.05%	-0.06%	2.42%	3.51%
	c	Base	1.3227	0.57%	0.53%	1.43%	2.10%
		Base+RI	1.3227	0.53%	0.39%	1.45%	2.46%
		Base+AS	1.3227	0.53%	0.27%	1.28%	2.31%

APPENDIX

C. ITEM PREDICTION

Table XIV. : Average Precision Performance for the MovieLens data set.

Data	User	Experimental	Recommendation List Size						
Subset	Expectations	Setting	*1*	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	0.74%	-4.04%	4.81%	9.84%	9.00%	6.09%	6.28%
		Homogeneous Quadratic	3.45%	0.08%	5.00%	7.62%	6.21%	5.01%	4.13%
		Heterogeneous Linear	-2.26%	-1.24%	4.59%	11.86%	12.73%	12.75%	14.49%
		Heterogeneous Quadratic	0.79%	1.40%	6.20%	12.86%	13.48%	13.46%	14.60%
	Base+RL	Homogeneous Linear	-1.79%	-4.35%	5.03%	10.45%	9.67%	6.71%	7.09%
		Homogeneous Quadratic	3.55%	0.18%	5.46%	7.96%	7.05%	5.57%	4.83%
		Heterogeneous Linear	-2.41%	-1.37%	4.96%	11.37%	12.11%	12.33%	14.39%
		Heterogeneous Quadratic	0.17%	1.27%	6.11%	11.90%	12.82%	12.99%	14.44%
b	Base	Homogeneous Linear	7.01%	5.81%	5.21%	4.81%	5.14%	4.57%	4.17%
		Homogeneous Quadratic	2.10%	3.65%	3.04%	2.71%	2.97%	2.34%	2.10%
		Heterogeneous Linear	4.24%	8.16%	8.88%	11.20%	11.37%	11.07%	12.07%
		Heterogeneous Quadratic	7.49%	8.77%	8.55%	9.46%	9.63%	9.01%	9.87%
	Base+RL	Homogeneous Linear	9.21%	6.93%	5.82%	5.26%	5.58%	4.87%	4.86%
		Homogeneous Quadratic	5.52%	5.55%	4.62%	4.01%	3.78%	3.22%	3.01%
		Heterogeneous Linear	2.91%	7.16%	7.64%	9.91%	10.28%	10.13%	11.47%
		Heterogeneous Quadratic	5.23%	6.89%	6.95%	8.42%	8.84%	8.18%	9.28%

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

Table XV. : Average Recall Performance for the MovieLens data set.

Data	User	Experimental	Recommendation List Size						
Subset	Expectations	Setting	*1*	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	2.02%	-0.40%	7.89%	15.11%	12.44%	9.45%	9.08%
		Homogeneous Quadratic	5.05%	3.95%	7.32%	11.94%	8.70%	7.44%	6.45%
		Heterogeneous Linear	3.03%	7.51%	13.24%	23.13%	22.26%	23.02%	25.30%
		Heterogeneous Quadratic	4.04%	9.49%	15.49%	25.56%	25.19%	25.32%	26.50%
	Base+RL	Homogeneous Linear	-2.02%	-1.19%	7.32%	15.86%	13.64%	10.43%	10.52%
		Homogeneous Quadratic	3.03%	3.16%	7.04%	12.87%	11.17%	9.35%	8.45%
		Heterogeneous Linear	3.03%	7.51%	12.39%	20.71%	20.61%	21.25%	24.13%
		Heterogeneous Quadratic	1.01%	9.88%	13.80%	21.46%	22.19%	22.53%	24.47%
b	Base	Homogeneous Linear	8.57%	8.87%	7.02%	7.05%	7.05%	5.85%	5.34%
		Homogeneous Quadratic	4.76%	5.80%	4.68%	4.23%	4.35%	3.44%	3.11%
		Heterogeneous Linear	7.62%	15.02%	15.53%	17.27%	16.25%	16.11%	18.02%
		Heterogeneous Quadratic	12.38%	15.70%	15.74%	16.22%	15.22%	14.45%	15.79%
	Base+RL	Homogeneous Linear	8.57%	9.56%	7.87%	7.76%	7.36%	6.12%	5.90%
		Homogeneous Quadratic	6.67%	7.85%	6.60%	5.76%	5.25%	4.46%	4.17%
		Heterogeneous Linear	6.67%	14.33%	14.68%	16.22%	15.13%	14.86%	17.52%
		Heterogeneous Quadratic	8.57%	13.65%	14.04%	15.39%	13.82%	12.88%	14.66%

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

Table XVI. : Average Precision Performance for the BookCrossing data set.

Data	**User**	**Experimental**	**Recommendation List Size**						
Subset	**Expectations**	**Setting**	*1*	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	40.00%	23.47%	23.91%	18.18%	12.50%	8.00%	7.69%
		Homogeneous Quadratic	41.25%	28.57%	28.26%	19.32%	13.75%	9.33%	7.69%
		Heterogeneous Linear	45.00%	22.45%	23.91%	15.91%	10.00%	6.67%	6.15%
		Heterogeneous Quadratic	37.50%	24.49%	26.09%	14.77%	10.00%	6.67%	6.15%
	Base+RI	Homogeneous Linear	66.25%	25.51%	27.17%	20.45%	15.00%	9.33%	9.23%
		Homogeneous Quadratic	46.25%	28.57%	27.17%	19.32%	15.00%	9.33%	9.23%
		Heterogeneous Linear	48.75%	28.57%	29.35%	18.18%	11.25%	9.33%	7.69%
		Heterogeneous Quadratic	33.75%	24.49%	26.09%	15.91%	10.00%	6.67%	6.15%
	Base+AR	Homogeneous Linear	33.75%	16.33%	17.39%	12.50%	8.75%	5.33%	6.15%
		Homogeneous Quadratic	36.25%	24.49%	21.74%	13.64%	10.00%	6.67%	7.69%
		Heterogeneous Linear	27.50%	19.39%	19.57%	10.23%	6.25%	4.00%	4.62%
		Heterogeneous Quadratic	31.25%	24.49%	23.91%	13.64%	7.50%	5.33%	4.62%
b	Base	Homogeneous Linear	41.36%	33.33%	21.00%	2.42%	5.29%	3.30%	1.06%
		Homogeneous Quadratic	46.07%	29.94%	20.50%	5.24%	5.73%	3.30%	1.06%
		Heterogeneous Linear	43.46%	32.77%	21.00%	2.02%	2.20%	1.89%	2.12%
		Heterogeneous Quadratic	40.31%	29.94%	18.50%	2.82%	2.64%	0.94%	0.53%
	Base+RI	Homogeneous Linear	49.21%	48.02%	32.00%	6.85%	7.05%	4.72%	3.17%
		Homogeneous Quadratic	47.12%	37.85%	28.00%	6.45%	6.61%	4.25%	2.65%
		Heterogeneous Linear	46.60%	44.07%	26.50%	5.65%	4.85%	2.83%	2.65%
		Heterogeneous Quadratic	48.17%	32.20%	19.50%	3.63%	2.64%	0.94%	0.53%
	Base+AR	Homogeneous Linear	38.74%	33.33%	25.00%	4.03%	3.96%	2.36%	1.59%
		Homogeneous Quadratic	39.27%	33.33%	24.50%	4.44%	4.41%	2.36%	1.59%
		Heterogeneous Linear	38.22%	31.64%	20.00%	1.61%	1.32%	0.00%	0.00%
		Heterogeneous Quadratic	43.98%	31.07%	18.00%	2.82%	1.76%	0.47%	0.00%
c	Base	Homogeneous Linear	10.99%	1.73%	0.93%	-1.14%	-0.19%	0.20%	0.20%
		Homogeneous Quadratic	13.19%	2.50%	1.67%	1.14%	0.58%	0.59%	0.39%
		Heterogeneous Linear	-1.54%	-3.08%	3.33%	2.84%	1.16%	1.97%	1.38%
		Heterogeneous Quadratic	-0.44%	-3.08%	0.93%	1.52%	0.39%	0.98%	0.59%
	Base+RI	Homogeneous Linear	11.65%	2.12%	-2.04%	-2.65%	-0.77%	0.39%	0.79%
		Homogeneous Quadratic	13.63%	5.19%	0.74%	0.19%	0.96%	1.18%	1.18%
		Heterogeneous Linear	-3.08%	-6.73%	-0.93%	-1.33%	-0.19%	1.57%	0.59%
		Heterogeneous Quadratic	1.10%	-3.65%	0.56%	0.57%	0.00%	0.59%	0.39%
	Base+AR	Homogeneous Linear	9.45%	-4.42%	-3.70%	-4.17%	-1.35%	-0.39%	0.20%
		Homogeneous Quadratic	14.07%	-2.31%	-2.04%	-1.70%	-0.58%	-0.20%	0.39%
		Heterogeneous Linear	-1.54%	-8.85%	-2.22%	-1.89%	-1.16%	0.39%	-0.20%
		Heterogeneous Quadratic	1.10%	-4.81%	0.19%	0.57%	-0.19%	0.59%	0.39%

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

Table XVII. : Average Recall Performance for the BookCrossing data set.

Data	**User**	**Experimental**	**Recommendation List Size**						
Subset	**Expectations**	**Setting**	*1*	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	36.00%	18.82%	24.41%	21.21%	13.13%	7.27%	7.08%
		Homogeneous Quadratic	36.00%	23.53%	26.77%	22.08%	14.26%	8.91%	7.79%
		Heterogeneous Linear	44.00%	20.00%	25.98%	20.35%	12.97%	8.91%	6.84%
		Heterogeneous Quadratic	48.00%	23.53%	30.71%	21.21%	12.97%	8.40%	6.01%
	Base+RI	Homogeneous Linear	52.00%	17.65%	27.56%	21.21%	14.59%	8.50%	9.10%
		Homogeneous Quadratic	40.00%	20.00%	25.98%	19.91%	15.40%	8.91%	9.76%
		Heterogeneous Linear	48.00%	25.88%	33.86%	22.51%	13.61%	10.04%	7.79%
		Heterogeneous Quadratic	44.00%	24.71%	31.50%	21.21%	12.48%	8.30%	5.83%
	Base+AR	Homogeneous Linear	36.00%	15.29%	21.26%	14.72%	10.05%	5.53%	7.08%
		Homogeneous Quadratic	40.00%	20.00%	22.05%	16.02%	11.67%	7.07%	8.74%
		Heterogeneous Linear	40.00%	22.35%	26.77%	17.32%	8.91%	6.97%	4.76%
		Heterogeneous Quadratic	44.00%	24.71%	29.92%	19.48%	11.18%	7.48%	5.23%
b	Base	Homogeneous Linear	35.00%	31.40%	20.63%	3.16%	6.36%	3.72%	1.84%
		Homogeneous Quadratic	37.50%	29.75%	20.18%	5.20%	6.81%	3.97%	1.93%
		Heterogeneous Linear	45.00%	35.54%	22.87%	3.90%	3.92%	2.02%	2.09%
		Heterogeneous Quadratic	42.50%	33.06%	20.18%	4.46%	3.98%	1.13%	1.36%
	Base+RI	Homogeneous Linear	50.00%	43.80%	30.94%	8.36%	6.81%	4.25%	3.25%
		Homogeneous Quadratic	42.50%	33.88%	27.35%	6.32%	7.39%	3.76%	2.85%
		Heterogeneous Linear	50.00%	39.67%	27.80%	8.36%	6.55%	2.59%	2.67%
		Heterogeneous Quadratic	50.00%	33.06%	20.18%	4.65%	3.79%	1.30%	1.06%
	Base+AR	Homogeneous Linear	37.50%	33.88%	23.32%	4.65%	4.11%	2.27%	1.93%
		Homogeneous Quadratic	32.50%	32.23%	22.87%	3.90%	5.33%	2.39%	2.00%
		Heterogeneous Linear	40.00%	29.75%	19.28%	3.72%	3.47%	0.12%	0.16%
		Heterogeneous Quadratic	47.50%	33.06%	19.28%	5.02%	3.34%	0.89%	0.58%
c	Base	Homogeneous Linear	17.14%	1.01%	-1.05%	-1.59%	-0.89%	0.65%	0.25%
		Homogeneous Quadratic	15.71%	2.03%	0.47%	1.16%	0.09%	0.90%	0.41%
		Heterogeneous Linear	-5.00%	-5.07%	1.05%	3.49%	0.28%	2.09%	1.56%
		Heterogeneous Quadratic	-6.43%	-3.85%	-1.99%	2.27%	-0.39%	1.24%	0.81%
	Base+RI	Homogeneous Linear	22.14%	2.64%	-3.39%	-3.56%	-2.10%	0.92%	0.74%
		Homogeneous Quadratic	19.29%	4.87%	-0.70%	-0.86%	-0.19%	1.68%	1.27%
		Heterogeneous Linear	-4.29%	-5.48%	-1.87%	0.00%	-1.14%	2.22%	1.13%
		Heterogeneous Quadratic	-2.14%	-4.46%	-1.52%	1.53%	-1.38%	0.50%	0.57%
	Base+AR	Homogeneous Linear	16.43%	-3.65%	-6.32%	-5.76%	-2.46%	-0.38%	0.03%
		Homogeneous Quadratic	15.00%	-3.04%	-4.44%	-3.49%	-1.84%	-0.13%	0.20%
		Heterogeneous Linear	-7.86%	-10.55%	-4.56%	-1.29%	-2.89%	0.53%	0.23%
		Heterogeneous Quadratic	-3.57%	-5.48%	-2.46%	0.92%	-1.69%	0.28%	0.28%

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 XVIII. : Average Coverage Performance for the MovieLens data set.

Data Subset	**User Expectations**	**Experimental Setting**	**Recommendation List Size**						
			1	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	59.65%	52.33%	46.56%	37.46%	21.66%	16.72%	10.65%
		Homogeneous Quadratic	59.14%	52.33%	46.91%	37.86%	22.09%	16.98%	10.64%
		Heterogeneous Linear	78.04%	70.75%	65.00%	56.14%	39.06%	34.27%	29.15%
		Heterogeneous Quadratic	75.08%	66.35%	60.32%	51.06%	33.65%	28.46%	22.07%
	Base+RL	Homogeneous Linear	61.49%	53.27%	47.64%	37.86%	21.64%	16.93%	11.47%
		Homogeneous Quadratic	61.39%	53.27%	47.37%	37.61%	21.82%	17.10%	11.51%
		Heterogeneous Linear	85.80%	76.87%	69.57%	59.19%	41.52%	36.61%	31.04%
		Heterogeneous Quadratic	82.43%	73.05%	66.12%	55.09%	37.13%	32.18%	26.14%
b	Base	Homogeneous Linear	24.04%	25.97%	26.56%	21.40%	12.13%	9.02%	5.40%
		Homogeneous Quadratic	24.18%	25.02%	25.40%	21.06%	12.41%	8.99%	5.53%
		Heterogeneous Linear	44.68%	45.84%	48.41%	43.13%	31.36%	27.57%	25.53%
		Heterogeneous Quadratic	37.23%	39.86%	41.43%	35.43%	23.94%	19.46%	16.81%
	Base+RL	Homogeneous Linear	25.18%	25.86%	27.08%	22.12%	13.21%	10.16%	6.75%
		Homogeneous Quadratic	24.26%	25.21%	26.18%	21.80%	13.36%	10.40%	7.70%
		Heterogeneous Linear	47.94%	52.56%	55.05%	49.68%	37.23%	33.09%	30.26%
		Heterogeneous Quadratic	43.12%	46.45%	48.90%	43.92%	32.80%	27.98%	25.38%

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

Table XIX. : Average Coverage Performance for the BookCrossing data set.

Data	**User**	**Experimental**	**Recommendation List Size**						
Subset	**Expectations**	**Setting**	*1*	*3*	*5*	*10*	*30*	*50*	*100*
a	Base	Homogeneous Linear	71.09%	44.62%	33.61%	21.65%	11.07%	8.04%	5.17%
		Homogeneous Quadratic	70.57%	44.36%	33.47%	21.75%	11.32%	8.42%	5.76%
		Heterogeneous Linear	85.23%	56.59%	43.93%	30.64%	18.91%	16.80%	17.25%
		Heterogeneous Quadratic	83.74%	58.34%	45.36%	30.98%	17.05%	13.30%	10.52%
	Base+RI	Homogeneous Linear	48.30%	29.38%	21.85%	14.48%	7.55%	5.50%	3.23%
		Homogeneous Quadratic	48.03%	29.21%	21.36%	14.06%	7.12%	5.02%	2.70%
		Heterogeneous Linear	79.10%	52.70%	40.85%	28.01%	16.04%	13.18%	11.60%
		Heterogeneous Quadratic	77.00%	50.77%	38.85%	25.73%	12.81%	9.13%	6.14%
	Base+AR	Homogeneous Linear	50.53%	31.04%	23.16%	15.36%	8.61%	6.42%	3.45%
		Homogeneous Quadratic	49.87%	30.22%	22.34%	14.57%	7.85%	5.62%	2.70%
		Heterogeneous Linear	80.05%	53.29%	41.47%	28.48%	16.13%	13.16%	11.34%
		Heterogeneous Quadratic	78.22%	51.20%	39.11%	26.02%	12.69%	8.92%	5.86%
b	Base	Homogeneous Linear	61.92%	39.92%	29.95%	18.27%	6.65%	4.28%	2.18%
		Homogeneous Quadratic	61.30%	39.36%	29.55%	18.02%	6.37%	4.08%	1.98%
		Heterogeneous Linear	71.48%	50.00%	39.25%	27.73%	18.21%	18.49%	22.05%
		Heterogeneous Quadratic	66.64%	44.64%	33.98%	21.78%	9.45%	7.82%	7.55%
	Base+RI	Homogeneous Linear	46.72%	31.29%	23.89%	13.83%	4.35%	2.68%	2.22%
		Homogeneous Quadratic	46.32%	31.56%	23.71%	13.79%	3.97%	1.79%	0.82%
		Heterogeneous Linear	79.88%	58.98%	48.33%	35.30%	24.74%	23.94%	25.35%
		Heterogeneous Quadratic	75.47%	54.60%	43.34%	29.65%	16.91%	14.68%	14.37%
	Base+AR	Homogeneous Linear	45.29%	29.61%	22.34%	12.54%	2.88%	0.97%	-0.10%
		Homogeneous Quadratic	45.02%	29.98%	22.35%	12.60%	2.91%	0.52%	-1.03%
		Heterogeneous Linear	77.21%	56.41%	45.65%	33.18%	22.26%	21.38%	22.92%
		Heterogeneous Quadratic	73.04%	52.18%	41.10%	27.57%	14.69%	12.24%	11.27%
c	Base	Homogeneous Linear	32.03%	19.68%	12.10%	5.48%	2.30%	1.91%	2.39%
		Homogeneous Quadratic	31.80%	19.18%	11.69%	5.37%	2.17%	1.81%	2.07%
		Heterogeneous Linear	41.57%	29.43%	21.96%	16.00%	15.10%	17.49%	20.61%
		Heterogeneous Quadratic	36.92%	22.72%	14.56%	6.99%	4.06%	4.66%	6.46%
	Base+RI	Homogeneous Linear	28.21%	17.66%	11.50%	7.30%	4.68%	5.76%	9.40%
		Homogeneous Quadratic	26.40%	15.35%	9.45%	4.54%	1.43%	2.74%	7.81%
		Heterogeneous Linear	53.70%	40.05%	32.11%	26.80%	24.53%	25.33%	25.52%
		Heterogeneous Quadratic	48.53%	33.56%	25.42%	18.66%	15.37%	15.82%	18.81%
	Base+AR	Homogeneous Linear	25.99%	13.91%	7.75%	3.54%	0.01%	0.45%	1.38%
		Homogeneous Quadratic	24.12%	11.41%	5.38%	0.71%	-4.26%	-4.40%	-4.39%
		Heterogeneous Linear	52.02%	38.13%	30.13%	23.65%	21.38%	22.38%	23.10%
		Heterogeneous Quadratic	46.93%	31.13%	22.68%	15.17%	10.26%	9.92%	10.37%

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

REFERENCES

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 2011)*. ACM. http://ceur-ws.org/Vol-816/paper2.pdf

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)*. http://ids.csom.umn.edu/faculty/gedas/NSFCareer/RS-WITS-2009-wp.pdf

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 2011)*. ACM. http://ceur-ws.org/Vol-816/paper1.pdf

G. Adomavicius and YoungOk Kwon. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *Knowledge and Data Engineering, IEEE Transactions on* 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 Ssystems: A Survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.* 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 Trans. Manage. Inf. Syst.* 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. http://ir.ii.uam.es/prsat2010/papers/paper1.pdf

Amazon 2012. Amazon.com, Inc. (2012). http://www.amazon.com 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, USA, 2033–2036. DOI:http://dx.doi.org/10.1145/1518701.1519009

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

Yoav Benjamini. 1988. Opening the Box of a Boxplot. *The American Statistician* 42, 4 (1988), pp. 257–262. http://www.jstor.org/stable/2685133

Gideon Berger and Alexander Tuzhilin. 1998. Discovering unexpected patterns in temporal data using temporal logic. *Temporal Databases: research and practice* (1998), 281–309.

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

Daniel Billsus and Michael J. Pazzani. 2000. User Modeling for Adaptive News Access. *User Modeling and User-Adapted Interaction* 10, 2-3 (Feb. 2000), 147–180. DOI:http://dx.doi.org/10.1023/A:1026501525781

BookCrossing 2004. BookCrossing, Inc. (2004). http://www.bookcrossing.com 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. *Manage. Sci.* 57, 8 (Aug. 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. *Manage. Sci.* 49, 11 (Nov. 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 (Nov. 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 2011)*. ACM.

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 2011) at the 33rd European Conference on Information Retrieval (ECIR 2011)*.

Òscar Celma and Perfecto Herrera. 2008. A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems (RecSys '08)*. ACM, 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. *The Journal of Industrial Economics* 39, 4 (1991), pp. 383–390. http://www.jstor.org/stable/2098438

R. Davidson and J.G. MacKinnon. 2004. *Econometric Theory and Methods*. Oxford University Press. http://books.google.com/books?id=vqkap8AwAy0C

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

M. Ge, D. Jannach, F. Gedikli, and M. Hepp. 2012. EFFECTS OF THE PLACEMENT OF DIVERSE ITEMS IN RECOMMENDATION LISTS. In *Proceedings of 14th International Conference on Enterprise Information Systems (ICEIS 2012)*.

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. (2012). http://books.google.com http://books.google.com.

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

GroupLens 2011. GroupLens Research group. (2011). http://www.grouplens.org http://www.grouplens.org.

Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 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, USA, 67–76. DOI:http://dx.doi.org/10.1145/1502650.1502663

Edgar Hoover. 1985. *An introduction to regional economics*. A. A. Knopf, New York.

R. Hu and P. Pu. 2011. Helping Users Perceive Recommendation Diversity. *Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011)* (2011), 43.

N. Hurley and M. Zhang. 2011. Novelty and Diversity in Top-N Recommendation–Analysis and Evaluation. *ACM Transactions on Internet Technology (TOIT)* 10, 4 (2011), 14.

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 2008 8th International Conference on Hybrid Intelligent Systems (HIS '08)*. IEEE Computer Society, Washington, DC, USA, 168–173. DOI:http://dx.doi.org/10.1109/HIS.2008.25

IMDb 2011. IMDb.com, Inc. (2011). http://www.imdb.com http://www.imdb.com.

ISBNdb.com 2012. The ISBN database. (2012). http://isbndb.com 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, USA, 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 third ACM conference on Recommender systems (RecSys '09)*. ACM, New York, NY, USA, 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 *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*. 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, USA, 1630–1633. http://dl.acm.org/citation.cfm?id=1597348.1597458

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 Trans. Knowl. Discov. Data* 4, 1, Article 1 (Jan. 2010), 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 (Aug. 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, USA, 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. (2012). http://www.librarything.com http://www.librarything.com.

J.S. Long. 1997. *Regression models for categorical and limited dependent variables*. Vol. 7. Sage Publications, Incorporated.

M. O. Lorenz. 1905. Methods of Measuring the Concentration of Wealth. *Publications of the American Statistical Association* 9, 70 (1905), pp. 209–219. http://www.jstor.org/stable/2276207

Alfred Marshall. 1920. *Principles of Economics*. Vol. 1. Macmillan and Co., London, UK. 611 pages.

John F. McDonald and Robert A. Moffitt. 1980. The Uses of Tobit Analysis. *The Review of Economics and Statistics* 62, 2 (1980), pp. 318–321. http://www.jstor.org/stable/1924766

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, USA, 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, UK, 219–233. http://dl.acm.org/citation.cfm?id=646180.682439

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, Heidelberg, 40–46. http://dl.acm.org/citation.cfm?id=1788314.1788320

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, USA, 949–958. DOI:http://dx.doi.org/10.1145/1871437.1871558

Damien Neven. 1985. Two Stage (Perfect) Equilibrium in Hotelling's Model. *The Journal of Industrial Economics* 33, 3 (1985), pp. 317–325. http://www.jstor.org/stable/2098539

Randall J. Olsen. 1978. Note on the Uniqueness of the Maximum Likelihood Estimator for the Tobit Model. *Econometrica* 46, 5 (1978), pp. 1211–1215. http://www.jstor.org/stable/1911445

Balaji Padmanabhan and Alexander Tuzhilin. 1998. A Belief-Driven Method for Discovering Unexpected Patterns. In *Proceedings of the third International Conference on Knowledge Discovery and Data Mining (KDD '98)*. AAAI Press, Palo Alto, CA, USA, 94–100.

Balaji Padmanabhan and Alexander Tuzhilin. 2000. Small is beautiful: discovering the minimal set of unexpected patterns. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '00)*. ACM, New York, NY, USA, 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 Trans. on Knowl. and Data Eng.* 18, 2 (Feb. 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 third ACM conference on Recommender systems (RecSys '09)*. ACM, New York, NY, USA, 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(21). http://www.stata-journal.com/article.html?article=st0005

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. http://research.microsoft.com/pubs/115396/EvaluationMetrics.TR.pdf

Yue Shi, Xiaoxue Zhao, Jun Wang, Martha Larsona, and Alan Hanjalic. 2012. Adaptive Diversification of Recommendation Results via Latent Factor Portfolio. In *SIGIR*.

P.K. Shivaswamy, Wei Chu, and M. Jansche. 2007. A Support Vector Approach to Censored Targets. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. 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. *Knowledge and Data Engineering, IEEE Transactions on* 8, 6 (dec 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, USA, 307–310. DOI:http://dx.doi.org/10.1145/1998076.1998133

Jean Tirole. 1988. *The Theory of Industrial Organization*. Mit Press. http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=8224

Akhmed Umyarov and Alexander Tuzhilin. 2011. Using external aggregate ratings for improving individual recommendations. *ACM Trans. Web* 5, 1, Article 3 (Feb. 2011), 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 fifth ACM conference on Recommender systems (RecSys '11)*. ACM, New York, NY, USA, 109–116. DOI:http://dx.doi.org/10.1145/2043932.2043955

S. Vargas, P. Castells, and D. Vallet. 2012. Explicit Relevance Models in Intent-Oriented Information Retrieval Diversification. In *35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012)*. Portland, OR, USA. http://ir.ii.uam.es/predict/pubs/sigir12-vargas.pdf

Jun Wang and Jianhan Zhu. 2009. Portfolio Theory of Information Retrieval. In *Proc. of the Annual International ACM SIGIR Conference on Research and Development on Information Retrieval (SIGIR)*. http://web4.cs.ucl.ac.uk/staff/jun.wang/blog/2009/06/17/portfolio-theory-of-information-retrieval/

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, USA, 115–118. http://dl.acm.org/citation.cfm?id=1339264.1339672

Wikipedia 2012. Wikimedia Foundation, Inc. (2012). http://www.wikipedia.org http://www.wikipedia.org.

J.M. Wooldridge. 2002. *Econometric Analysis of Cross Section and Panel Data*. Mit Press. http://books.google.com/books?id=cdBPOJUP4VsC

WorldCat 2012. OCLC Online Computer Library Center, Inc. (2012). http://www.worldcat.org 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, 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, USA, 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 fifth ACM international conference on Web search and data mining (WSDM '12)*. ACM, New York, NY, USA, 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, USA, 22–32. DOI:http://dx.doi.org/10.1145/1060745.1060754