

# Classical Music for Rock Fans?: Novel Recommendations for Expanding User Interests

Makoto Nakatsuji  
NTT Cyber Solutions  
Laboratories, NTT Corporation  
nakatsuji.makoto  
@lab.ntt.co.jp

Toshio Uchiyama  
NTT Cyber Solutions  
Laboratories, NTT Corporation  
uchiyama.toshio  
@lab.ntt.co.jp

Yasuhiro Fujiwara  
NTT Cyber Space  
Laboratories, NTT Corporation  
fujiwara.yasuhiro  
@lab.ntt.co.jp

Ko Fujimura  
NTT Cyber Solutions  
Laboratories, NTT Corporation  
fujimura.ko  
@lab.ntt.co.jp

Akimichi Tanaka  
NTT Cyber Solutions  
Laboratories, NTT Corporation  
tanaka.akimichi  
@lab.ntt.co.jp

Toru Ishida  
Department of Social  
Informatics, Kyoto University  
ishida@i.kyoto-u.ac.jp

## ABSTRACT

Most recommender algorithms produce types similar to those the active user has accessed before. This is because they measure user similarity only from the co-rating behaviors against items and compute recommendations by analyzing the items possessed by the users most similar to the active user. In this paper, we define item novelty as the smallest distance from the class the user accessed before to the class that includes target items over the taxonomy. Then, we try to accurately recommend highly novel items to the user. First, our method measures user similarity by employing items rated by users and a taxonomy of items. It can *accurately* identify *many* items that may suit the user. Second, it creates a graph whose nodes are users; weighted edges are set between users according to their similarity. It analyzes the user graph and extracts users that are related on the graph though the similarity between the active user and each of those users is not high. The users so extracted are likely to have *highly novel items* for the active user. An evaluation conducted on several datasets finds that our method accurately identifies items with higher novelty than previous methods.

## Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning—*Knowledge Acquisition*; H3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Information Filtering*

## General Terms

Algorithms, Experimentation, Theory

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.  
Copyright 2010 ACM 978-1-4503-0099-5/10/10...\$10.00.

## Keywords

Collaborative filtering, Novelty Detection

## 1. INTRODUCTION

Recommender systems are widely used by content providers to increase their chance of commercial success. Many content providers adopt methods based on collaborative filtering (CF), which is a broad term for the process of recommending items to an active user, the one who is receiving the recommendation, based on the intuition that users who access the same items with the user tend to have similar interests to the user. Basic CF methods measure the similarity of users only from the co-rating behaviors against items, and compute recommendations for the active user by analyzing the items possessed by the users most similar to the user. As a result, they are apt to recommend the types of items that have already been accessed by the user. For example, if the user highly rates a horror movie (as an item), the typical CF methods recommend items that were made by the same director, performed by the same actors, or included in the same genre, horror. The user, however, may have variety of interests other than horror movie items and often needs recommendations of other types of items[25].

Against this problem, Ziegler and McNee proposed an algorithm that increases the diversity of recommendation list entries[25]. The list includes items in several classes defined by a taxonomy. This can raise user satisfaction but the accuracy of item prediction is degraded. Their recommendations consider only the items possessed by the users most similar to the user and so include the same types of items that the user has accessed before; they do, however, present a greater number of classes than traditional CF techniques. Their method also fails to analyze or measure item novelty, a critical omission since novel recommendations are essential to widening the user's range of interests. We assess novelty in terms of the distance from the present interests of the user to the item recommended according to the taxonomy of items. This metric is useful in two ways. First, it presents, in an easy to understand manner, the relationship between the user's present interests and the target item. That is, the user can understand why the presented items are different

from those that the user has accessed before. Second, the user can boost the novelty threshold if she wants items that are completely unknown to her.

In this paper, we propose a method that *accurately* identifies items that are *highly novel* for the active user. We define item novelty as the smallest distance from the class the user accessed before to the class that includes a target item. The metric of accuracy measures the performance of the recommender algorithm by comparing the algorithm’s prediction against a user’s rating of an item. This metric is also important because users more strongly trust accurate recommendation results and tend to use such services[12, 16]. By accurately identifying items that are highly novel to the user, and recommending those to him, he may become interested in them. For example, if the recommender system can recommend many good Classical music items to a user who has been listening to only Rock music items, the user may widen his interests by accessing the recommended items. The content providers also can promote more of their items to users even if the users previously accessed only limited sets of items.

The main contributions of the paper are the following.

- We introduce a new measure, *the score of novelty*, to understand *how novel* the recommended items are for the user. We define the score of novelty as the smallest distance from the class the user accessed before to the class that includes a target item over the taxonomy. By using this measure, the active user can understand *how far the recommended items are* from his present interests.
- To *accurately* identify *many* items for the user, we introduce the taxonomy-based approach. Each service provider carefully designs its item taxonomy to enable their customers to access their preferred items easily. Thus, we consider that users who like items, may like the classes that include those items. Our method thus reflects the rating of the user on an item to that of a class that includes that item. Next, it measures user similarity by using not only co-rating behaviors against items but also those against classes in the taxonomy. Our method focuses in particular on the overlaps of user interests in determining user similarity; that is, it checks how many subclasses are shared and not shared by users in each class in the taxonomy. As a result, we can accurately identify many items for the active user by analyzing the interests of users who share same items and/or same classes with the active user.
- To identify *items with higher novelty* for the active user, we employ the graph-based approach. Users who are *similar* to the active user tend to share the same items in the same classes with the user, and so are not likely to provide highly novel items for the user. We create a user graph whose nodes are users and that sets weighted edges between users according to their similarity. We perform Random Walk with Restarts (RWR)[10, 6, 21] on the user graph, and try to extract user nodes that the walk arrives at frequently even though the weights on the edges from the starting node (active user node) to those extracted user nodes are small; this means that such users share few classes with the active user. Thus, we incorporate items held

by the users so discovered to compute the prediction values for the active user to identify items with higher novelty.

We evaluated the accuracy and novelty of our method comprehensively by using four large datasets of various item domains with many parameter settings. Furthermore, we extend the taxonomy of items in one item provider by applying the metadata of items gathered from another item provider, and evaluated how the enhanced taxonomy influenced the performance of our method. From the many evaluation results so gained, we confirm that our method identifies many more items that offer higher novelty and does so more accurately than previous similarity schemes including the taxonomy-based method proposed by Ziegler et al.[24, 25].

The paper is organized as follows: we describe related works in the next section. Section 3 explains the background of our study. Section 4 introduces how to measure the similarity of users following the taxonomy of items and identify items with higher novelty from a user graph created based on the similarity of users. Section 5 evaluates our method using several datasets, and finally, section 6 concludes the paper.

## 2. RELATED WORKS

In this section, we briefly review related works, which can be categorized into three parts: novel item recommendation, accurate item recommendation, and cross-domain recommendation.

**Novel Item Recommendation.** The notion of *novelty* is defined in different ways in several papers. To the best of our knowledge, most papers refer to the definition provided in [4]. They define novel items as items that are unknown to the user but interesting for the user. They also describe that novel items and serendipitous items are different in that the former is more easily found by the user than the latter. However, this definition is very abstract, which makes it difficult to evaluate item novelty in detail and thus apply it. We consider that highly novel items, based on our definition, are those that cannot be easily discovered by the user. For example, the user who, up to now, has demonstrated an interest only in music items in Rock, is unable to easily discover interesting items in Classical music by himself.

Onuma et al. proposed a method that identifies novel items as items (they call ‘surprising’) that are accessed by users similar to the active user but also accessed by users not so similar to the active user[15]. Their idea is to envision the problem of identifying novel items as node selection on a graph. They give high scores to nodes that are well connected to the older choices, and at the same time well connected to unrelated choices. Their evaluation example shows that their method generates more diverse recommendation results such as recommending ‘surprising’ items in comedy, horror, and science fiction movie items to the user who has shown an interest in only comedy movies. We consider that the score of novelty in our paper is more a natural definition because it allows item taxonomy to be used in explaining to the active user why the items were recommended as novel.

The authors in [19, 20] define novelty as “the degree to which a recommendation is non-obvious”. They propose the measurement of novelty that involves computing the number

of correct recommendations made by the algorithm that are not present in the recommendations made by a related algorithm. Their evaluation shows that their proposed method can identify items that can not be identified by the method that only ranks items by their popularity. We however consider such a metric reflects only popularity and is not suitable for evaluating novelty.

As explained above, we consider that the definitions of novelty given by previous works are not useful in helping the active user to understand why the recommended items were determined to be *novel* and to understand *how novel they are*.

Different from the above papers, our previous work examined user satisfaction when recommending novel items to the user[14]. We proposed a taxonomy-based algorithm to find *novel items* that we defined as items that are included in classes that the active user has not accessed before. An online evaluation showed that user clicks tend to concentrate on the novel items thus found. Our previous work, however, only focused on user interest analysis from the descriptions submitted to blogs and the application of novel item recommendation for blog community activation. It also considered only the items possessed by the users most similar to the user and so its recommendations are apt to include a few items with high novelty.

**Accurate Item Recommendation.** In most CF studies, the researchers focus on improving the accuracy of the prediction results. Here, we explain some of those works such as those using the matrix factorization technique, taxonomy-based technique, and graph mining technique.

Yehuda et al. proposed a method that uses matrix factorization. It characterizes both items and users by vectors of factors inferred from item rating patterns[7, 8]. High correspondence between item and user factors leads to a recommendation. This approach has become popular in recent years since it offers good scalability with predictive accuracy. However, their method does not aim to suggest novel items for the user. Furthermore, the matrix factorization technique usually analyzes latent factors inferred from item rating patterns, thus it is not easy for a user to understand why the identified items are being recommended to the user. We consider that presenting semantic reasons to the user is important, especially when recommending novel items. Thus, we use the taxonomy of items in explaining the reasoning behind the recommendations.

Some researchers use item taxonomies to raise the accuracy of prediction results[24]. This approach was shown to be useful when the transaction data of users is sparse. However, in measuring user similarity, this type of method focuses only on classes that include items rated by both users and their super classes. As a result, this method naively assumes that users who share many items are highly similar to the user; those users may have many good as well as many not so good items for the user. Our method is different from existing taxonomy-based methods because it assesses how many subclasses are shared and not shared in each class in the taxonomy.

The authors in [19, 20] assign a-priori scores to the classes in the taxonomy of items, and compute the relationships between scores assigned to different classes. They then propagate those scores for a specific user to predict each user's preference. Their method was shown to be useful when the transaction data of users was sparse. Their method lies out-

side the scope of CF methods (they call their approach ontology filtering) because they do not compute user similarities. Unfortunately, their method is not suitable for realizing highly novel items identification because the score of user interests assigned to one class is rarely propagated to the class that is far from the original class over the taxonomy.

Some researchers have started to use random walks or RWR on a graph to compute recommendations[23, 3, 6, 2, 15]. Yildirim and Krishnamoorthy perform random walks on the item graph whose nodes on the graph are items and whose weighted edges are set between item nodes according to item similarity; they confirmed that their approach overcame the sparsity problem, which decreases prediction accuracy when the transaction dataset is sparse. However, to the best of our knowledge, no study has identified items with higher novelty using random walks on the user graph.

**Cross-domain Recommendation.** Related to the studies of novel item identification, a few recent works have targeted cross-domain recommendation[9, 13], i.e. the prediction of items that are located in the domains that the active user explicitly did not show an interest in before. Bin and his co-workers analyzed users who took similar rating behaviors against items across several item domains[9]. Their method shares the knowledge that is learned by using the rating datasets from multiple item domains even when the users and items of these datasets do not overlap.

Nakatsuji et al. also proposed cross-domain recommendation based on the observation that users who share similar items or who share social connections, can provide recommendation chains (sequences of transitively associated edges) to items in other domains[13]. We consider, however, that novel item identification within a domain is still important because users who access items in a domain have already expressed an interest in that domain. Thus, it is natural to present novel items within the domain to expand the user's interests, by analyzing user interests in detail using a domain-specific taxonomy.

### 3. BACKGROUND

Our method extends CF and uses RWR to identify items with high novelty.

#### 3.1 Collaborative Filtering

CF methods can be classified into two approaches: memory-based CF and model-based CF. Memory-based CF assumes that each user belongs to a larger group of users with similar behavior. Indeed this method is referred to as user-oriented memory-based CF; an analogous method which builds item similarity groups using co-purchase history is known as item-oriented CF[18]. On the other hand, model-based CF generates the predictions by using a model that is optimized against training data. Clustering[5, 22] and Bayesian network models are examples of the model-based approach[6, 23].

In computing user similarity, basic CF methods often use the Pearson correlation approach[17] or the cosine-based approach[1].

If we define  $a$  as the active user, who is to receive the recommendation, and  $M$  as the set of items rated by users  $a$  and  $u$ ,  $r_{a,I_i}$  as the rating value of user  $a$  for item  $I_i$ , and  $\bar{r}_a$  as the average value of item ratings given by  $a$ , the Pearson correlation coefficient measures the similarity  $S(a, u)$  between  $a$  and  $u$  as indicated by equation (1).

$$S(a, u) = \frac{\sum_{I_i \in M} (r_{a, I_i} - \bar{r}_a)(r_{u, I_i} - \bar{r}_u)}{\sqrt{\sum_{I_i \in M} (r_{a, I_i} - \bar{r}_a)^2} \sqrt{\sum_{I_i \in M} (r_{u, I_i} - \bar{r}_u)^2}} \quad (1)$$

When we use the cosine-based approach, we set  $\bar{r}_a$  and  $\bar{r}_u$  as zero in equation (1). The advantage of the Pearson correlation approach is that it takes into account that different users might have different rating schemes.

If we assume  $N$  is the set of users that are most similar to  $a$ , the predicted rating of  $a$  on item  $I_i$ ,  $p_{a, I_i}$  is obtained by the following equation (2).

$$p_{a, I_i} = \bar{r}_a + \frac{\sum_{u \in N} (r_{u, I_i} - \bar{r}_u) S(a, u)}{\sum_{u \in N} S(a, u)} \quad (2)$$

### 3.2 Random Walk with Restarts

A graph is a natural representation of data objects that have some inherent relational structure. In a graph, objects and their relationships can be represented as nodes and weighted edges respectively, where weights denote the strength of a relationship. Measuring the relatedness of two nodes in the graph can be achieved by using RWR theory[10]. Starting from node  $a$ , a RWR is performed by following a randomly selected link to another node at each step. Additionally, at every step there is a probability,  $\alpha$ , that the walk restarts at  $a$ . Let  $\mathbf{p}^{(t)}$  be a column vector where  $p_u^{(t)}$  denotes the probability that the random walk at step  $t$  proceeds from node  $u$ .  $\mathbf{q}$  is a column vector whose elements are set to zero; only the element corresponding to  $a$  is set to one, i.e.  $q(a) = 1$ . Also let  $\mathbf{A}$  be the column-normalized adjacency matrix of the graph. In other words,  $\mathbf{A}$  is the transition probability table where element  $A(u, v)$  gives the probability of  $v$  being the next node given that the current node is  $u$ . The stationary probabilities for each node can be obtained by recursively applying equation (3) until convergence; they give us the long-term visit rate of each node biased towards the starting node.

$$\mathbf{p}^{(t+1)} = (\mathbf{1} - \alpha)\mathbf{A}\mathbf{p}^{(t)} + \alpha\mathbf{q} \quad (3)$$

Therefore,  $p_a^{(l)}$ , where  $l$  is the status after convergence, can be considered as a measure of relatedness between nodes  $a$  and  $u$ .

## 4. PROPOSED METHOD

In this section, we detail the proposed method. First, we define the problem. We then introduce our method; it models user interests and measures user similarity according to the taxonomy of items. Next, we show how novel item identification can be realized by performing RWR on the user similarity graph.

### 4.1 Problem Definition

Table 1 gives the main symbols used in this paper.

We first define the novelty of an item for the active user using the taxonomy of items. We consider that it presents, in an easy to understand manner, the relationship between the user's present interests and the target item. Let the distance from class  $C_i$  to  $C_j$  over the taxonomy be  $dis(C_i, C_j)$  and  $\mathcal{I}_i(a)$  be the item set in class  $C_i$  that user  $a$  has rated. Then, our formal definition of the score of novelty  $\mathcal{N}(a, I_j)$  against

Table 1: Definition of main symbols.

Symbols	Definitions
$a$	The active user
$C_i$	A class
$I_i$	An item in $C_i$
$\mathcal{I}_i(u)$	The set of items in class $C_i$ that user $u$ rates
$\mathcal{C}_i(u)$	The set of subclasses of class $C_i$ that user $u$ rates
$r_{u, I_i}$	Rating value of $u$ against item $I_i$
$r_{u, C_i}$	Rating value of $u$ against class $C_i$
$dis(C_i, C_j)$	The distance from $C_i$ to $C_j$
$S(a, u, C_i)$	The score of interest agreement between users $a$ and $u$ against class $C_i$
$S_C(a, u)$	The similarity of rating behaviors against all classes between $a$ and $u$
$S_I(a, u)$	The similarity of rating behaviors against all items between $a$ and $u$
$S(a, u)$	The similarity of user $a$ and $u$
$\mathcal{N}(a, I_j)$	The score of novelty against item $I_j$ for user $a$

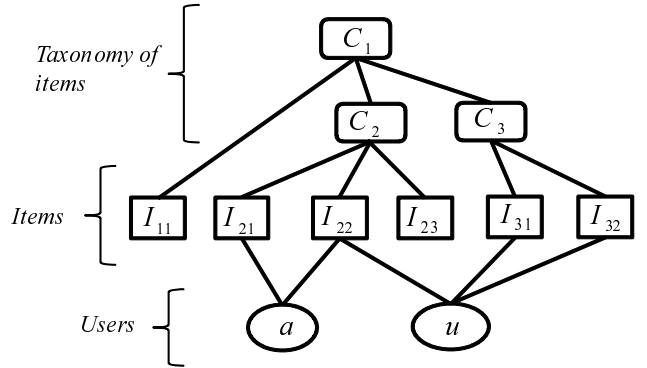


Figure 1: An explanation of the score of novelty.

item  $I_j$  in  $C_j$  for user  $a$ , who has already has expressed interest in  $I_i$  in  $C_i$ , as follows;

$$\mathcal{N}(a, I_j) = \min\{dis(C_i, C_j) | \mathcal{I}_i(a) \neq \emptyset\} \quad (4)$$

We assess the distance as the hops from class  $C_i$  to class  $C_j$  over the taxonomy. For example, in Fig. 1, if active user  $a$  accessed items  $I_{21}$  and  $I_{22}$  in class  $C_2$ , items  $I_{31}$  and  $I_{32}$  have novelty scores of two for  $a$  because there are two classes  $C_1$  and  $C_3$  between  $C_2$  and those items.

We define the problem tackled in the paper as follows:

**Problem** (RECOMMENDING HIGHLY NOVEL ITEMS).

**Given:** A taxonomy of items and ratings of users against items.

**Find:** Highly novel items for the user, while guaranteeing high prediction accuracy.

### 4.2 Modeling user interests

Taxonomies are becoming more readily available; examples include the taxonomies of music, movies, and game content generated by All Media Guide<sup>1</sup> and ListenJapan<sup>2</sup>.

<sup>1</sup><http://www.allmediaguide.com/>

<sup>2</sup><http://listen.jp/>

---

**Algorithm 1** Building user interests.

---

**Input:** A taxonomy of items and ratings of users against items.

**Output:** Interests of  $u$ .

```
1: //Measuring rating of  $u$  against each class  $C_i$ .
2: for each class  $C_i$  do
3:    $r_{u,C_i} = \sum_{I_i \in \mathcal{I}_i(u)} r_{u,I_i}$ ;
4: end for
5: //Reflecting ratings of subclasses to that of the super class.
6: for  $k = 1$  to  $H - 1$  do
7:   for each class  $C_j$  in  $L(H - k)$  do
8:     for each class  $C_l$  in  $C_j(u)$  do
9:        $r_{u,C_j} = \sum_{C_l \in C_j(u)} r_{u,C_l}$ ;
10:    end for
11:  end for
12: end for
```

---

Tabelog<sup>3</sup> has created a deep taxonomy for restaurants in Japan. We consider that modeling user interests according to these taxonomies is reasonable because content providers make significant efforts to optimize the granularity and branching factors of classes so that their customers can readily access the items according to the customers' interests. For example, the music taxonomy of ListenJapan has, on average, four hierarchies and 432 classes to classify items, while the restaurant taxonomy of Tabelog has, on average, four hierarchies and 318 classes to classify items.

Our approach is based on the observation that users who are interested in some items are also interested in classes that include those items. Thus, we reflect the rating values of the items to that of the class that includes those items. In the same way, we reflect the rating values of the classes to the super class that includes those classes. The rating value for an item is implicitly assigned according to the frequency of a user's access to the item, or explicitly assigned by the user.

The algorithm for building user interests is shown in Algorithm 1. We rate the class from the ratings of items in the class. Formally, the rating value of a class,  $r_{u,C_i}$ , is computed as  $\sum_{I_i \in \mathcal{I}_i(u)} r_{u,I_i}$  (line 3). For example in Fig.2, if user  $u$  assigns the rating value of 4.0 to  $I_{31}$  and 4.5 to  $I_{32}$ , the rating value of class  $C_3$  for  $u$  is 8.5. The rating values of the super class are computed in the same way; a key point is that we use the rating to each class, instead of the rating to each item, and compute the rating values of the classes in the order of the depth of hierarchy level in the taxonomy (lines 6 - 12). In Algorithm 1, integer  $H$  is the maximum value of hierarchy level of the taxonomy, and  $L(k)$  is a function that returns class set in the  $k$ th hierarchy level in the taxonomy. Here, we set the root class in the first hierarchy level in the taxonomy.

### 4.3 Measuring user similarity

Next, we explain the similarity measurement between users  $a$  and  $u$  according to the taxonomy of items. We first explain the approach of our method and then present an example. Please refer to Algorithm 2 also. In the approach described

<sup>3</sup><http://tabelog.com/>: Tabelog is one of the most famous online restaurant guides in Japan. The word 'tabelog' means 'eating' in Japanese.

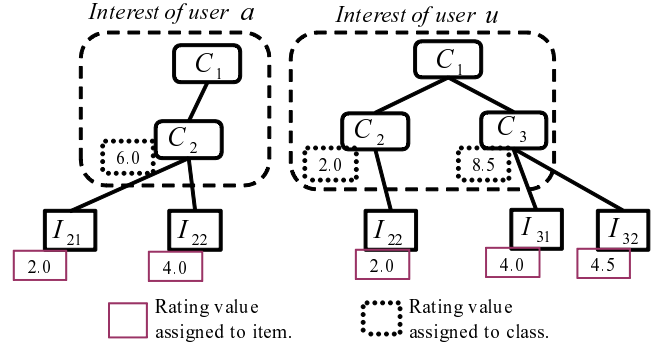


Figure 2: Measuring similarity of user  $a$  and  $u$ .

---

**Algorithm 2** Measuring user similarity

---

**Input:** Interests of users.

**Output:** Similarity scores between  $a$  and  $u$ .

```
1: for each user  $u$  do
2:   for each class  $C_j$  in  $\{C_i(a) \cup C_i(u)\}$  do
3:     compute  $S(a, u, C_j)$ ;
4:   end for
5:   Compute similarity scores  $S_C(a, u)$ ;
6:   Compute similarity scores  $S_I(a, u)$ ;
7:   Normalize  $S_C(a, u)$  and  $S_I(a, u)$ ;
8:   Compute  $S(a, u)$  as  $S_C(a, u) + S_I(a, u)$ ;
9: end for
```

---

below, we use  $\mathcal{C}_i(u)$  as the subclass set of class  $C_i$  that user  $u$  rates.

#### 4.3.1 Approach

- We first compute  $S(a, u, C_i)$ , the score of interest agreement between users  $a$  and  $u$  against class  $C_i$ . This rating value takes a smaller value in  $r_{a,C_i}$  and  $r_{u,C_i}$ ,  $\min(r_{a,C_i}, r_{u,C_i})$ . Thus we can filter out low-interest users through the score of interest agreement.
- Next, we compute the similarity of rating behaviors between  $a$  and  $u$  against all classes  $C$ , denoted as  $S_C(a, u)$ , with  $S(a, u, C_i)$  as follows:

$$S_C(a, u) = \sum_{C_i \in C} \frac{\sum_{C_j \in \{C_i(a) \cap C_i(u)\}} S(a, u, C_j)}{|C_i(a) \cup C_i(u)|} \quad (5)$$

The denominator utilizes the idea of the Jaccard coefficient[11] since it can effectively separate user  $u$  who assigns ratings against many classes from user  $a$  who assigns ratings to fewer classes. The Jaccard coefficient considers *not* co-rating classes as well as co-rating classes by utilizing the union of class sets. This is based on our observation that the interests of users are always not categorized as the same type even though users share many sub-classes. For example, readers can naturally guess that users who love Rock and Jazz genres are somewhat different from those who love Rock, Jazz, and Classic genres. By carefully analyzing such nuances in measuring user similarity, we can accurately

identify many items for the active user. The Pearson correlation approach and cosine-based approach do not have this property since they only consider the classes that are assigned ratings by both user  $a$  and  $u$  (see equation (1)).

- We then measure  $S_I(a, u)$ , the similarity of rating behaviors against items between users  $a$  and  $u$ . We use the Pearson correlation approach because it can handle the difference in the rating schemes of each user against items as explained in equation (1). Note that our approach can also employ the Jaccard coefficient approach.
- Finally, we normalize  $S_C(a, u)$  and  $S_I(a, u)$ , and combine the two above similarities,  $S_C(a, u)$  and  $S_I(a, u)$ , to evaluate the similarity of rating behaviors against classes and items between users.

The proposed method checks how many interests shared and not shared in each class in the taxonomy as well as offset the difference in the rating schemes of users. It is a natural approach and achieves high accuracy as demonstrated in our evaluation.

#### 4.3.2 Example

We explain our algorithm by using the example in Fig.2. (1)  $S(a, u, C_2)$  is computed as  $\min(6.0, 2.0) = 2.0$ .  $S(a, u, C_3)$  is computed as  $\min(0.0, 8.5) = 0.0$  (2)  $S_C(a, u)$  is computed as  $(2.0/2) + 0.0 = 1.0$ . (3) Next,  $S_I(a, u)$  is computed as  $\frac{(4-3)(2-3.5)}{\sqrt{(4-3)^2}\sqrt{(2-3.5)^2}} = -0.667$ . (4) Finally,  $S(a, u)$  is measured by combining the two above similarity scores,  $S_C(a, u)$  and  $S_I(a, u)$ .

### 4.4 Identifying highly novel items from the user graph

---

#### Algorithm 3 Predicting Items via Novelty Scores.

---

**Input:** A taxonomy of items and similarity scores between users.

**Output:** Prediction results with scores of novelty.

```

1: //Creating a column-normalized adjacency matrix A.
2: for each user  $u$  do
3:    $Sum(u) = 0$ ;
4:   for each user  $v$  do
5:      $Sum(u) += S(u, v)$ ;
6:   end for
7:    $A(u, v) = S(u, v) / Sum(u)$ ;
8: end for
9: Perform RWR on the graph determined by  $\mathbf{A}$  in finite
   times and get relatedness between  $a$  and  $u$ ;
10: Compute prediction values of items for  $a$  by applying
    relatedness instead of similarity using equation (2);
11: // Compute scores of novelty for all identified items.
12: for each item  $I_j$  predicted to  $a$  do
13:   Compute  $\mathcal{N}(a, I_j)$ ;
14: end for
```

---

We now explain how to combine RWR and similarity measurement results to identify *highly novel items* for users. Please refer to Algorithm 3.

Our method first builds a user graph whose nodes are users; the weighted edges between node  $u$  and  $v$  correspond

to the similarity score  $S(u, v)$ , see the previous subsection. In other words, we use  $S(u, v)$  to build column-normalized adjacency matrix  $\mathbf{A}$  in equation (3) by linearly scaling up each row of values such that the maximum of each row corresponds to 1 and the minimum of each row corresponds to 0 (lines 1 - 8).

Next, it executes RWR on the graph (line 9). By performing RWR on the graph, we obtain the probability that a walk from active user  $a$  will pass through user  $u$  on the graph; RWR, equation (3), is iterated until convergence is realized. Finally, we obtain the relatedness between active user node  $a$  and other users on the graph. In equation (3), as  $\alpha$  is decreased, the walk passes through more neighbors that diverge more strongly from active user  $a$ . We discuss the effect of this parameter in our evaluations.

By using *relatedness* instead of *similarity* between users  $a$  and  $u$ , we compute the prediction values of items for user  $a$  using equation (2) (line 10). Since our method incorporates items possessed by users who are not *similar* to user  $a$ , but who are *related* to  $a$  on the graph, it can recommend items with high novelty to  $a$ ; this means that such users share less items or less classes with the active user.

Then, we assign a score of novelty to each identified item following equation (4) (lines 11 - 14).

## 5. EVALUATION

We now evaluate our method for investigating the accuracy and item novelty yielded by our method.

### 5.1 Datasets

Our experiment used the following datasets.

#### 5.1.1 Ratings against movies

We used data from MovieLens<sup>4</sup>, an online movie recommendation system, to evaluate the performance of the proposed method. This data set contains 212,586 ratings of 943 anonymous MovieLens users on 1,682 movies. The average number of ratings assigned to each item is 59.4. Many online content providers do not have such a lot of number of ratings assigned to each item. For example, the average number of ratings assigned to each item is 4.64 in the dataset of Tabelog. The taxonomy of movie items is simple; it has only 19 genres as classes, and has only two hierarchies; the root class, “Movie” and subclasses. This dataset is much more dense than the music datasets explained later. Different from the implicit ratings extracted from blogs, a low item rating indicates a negative interest in that item. Because the number of classes is only 19 and the users rate many items in this dataset, thus this dataset is unsuitable for evaluating novel item identification. However, we can use this dataset to evaluate the accuracy of our method, even if the taxonomy is very simple and item ratings are dense.

The evaluation also uses the MovieLens dataset with an extended taxonomy enriched by the metadata of movie items provided by another movie information provider, goo movie<sup>5</sup>.

#### 5.1.2 Ratings against non-Japanese music artists

This dataset includes 48,695 implicit ratings of 3,545 users according to a taxonomy extracted in the experiment of

<sup>4</sup>Available at <http://www.grouplens.org/node/73>

<sup>5</sup><http://movie.goo.ne.jp/>

Table 2: MAE against movie dataset.

$T/D$	0.1			0.3			0.5			0.7		
$\alpha$	Non	0.8	0.6	Non	0.8	0.6	Non	0.8	0.6	Non	0.8	0.6
<i>Pearson</i>	1.07	1.06	1.06	0.90	0.90	0.89	0.84	0.84	0.84	0.83	0.83	0.83
<i>Cosine</i>	1.05	1.04	1.03	0.90	0.90	0.90	0.86	0.86	0.86	0.87	0.87	0.87
<i>Ziegler</i>	1.13	1.14	1.13	0.91	0.91	0.91	0.89	0.89	0.88	0.90	0.90	0.90
$T(J\&P)$	1.04	1.05	1.06	0.91	0.91	0.91	0.88	0.88	0.88	0.88	0.88	0.88
$T(J)$	0.97	0.97	0.96	0.88	0.88	0.88	0.86	0.86	0.86	0.87	0.87	0.87

Table 3: MAE against non-Japanese music dataset.

$T/D$	0.1			0.3			0.5			0.7		
$\alpha$	Non	0.8	0.6	Non	0.8	0.6	Non	0.8	0.6	Non	0.8	0.6
<i>Pearson</i>	1.36	1.41	1.41	1.26	1.23	1.22	1.20	1.17	1.15	1.23	1.22	1.23
<i>Cosine</i>	1.38	1.40	1.40	1.33	1.35	1.35	1.25	1.25	1.25	1.23	1.20	1.22
<i>Ziegler</i>	1.42	1.43	1.43	1.33	1.33	1.32	1.24	1.23	1.23	1.22	1.21	1.21
$T(J\&P)$	1.38	1.33	1.35	1.19	1.19	1.19	1.11	1.12	1.12	1.14	1.17	1.18
$T(J)$	1.39	1.37	1.35	1.22	1.22	1.22	1.13	1.14	1.14	1.15	1.17	1.18

Nakatsuji et al.[14] from the blog portal Doblog<sup>6</sup> and the taxonomy of non-Japanese music artists provided by ListenJapan. The taxonomy contains 279 genres as classes and 21,214 artists as items<sup>7</sup>. Nakatsuji et al. created rating values for each item of a user by analyzing the description frequency of each item among the user’s blog entries. The average number of ratings assigned to each item is 6.3. We linearly scaled up each rating value such that the maximum user rating corresponded to 5 and the minimum corresponded to 1 following the range of ratings in the MovieLens dataset. The class hierarchy in this taxonomy is deep; it has, on average, four hierarchies, and sometimes has a fifth hierarchy under the root class “Music” with detailed end classes such as “Space rock” and “Acid jazz”. This represents detailed expert knowledge that can be used to measure the similarity of users accurately. The greatest value in the score of novelty is six. Such large scores are assigned to recommended items if the items are under “Classic” class and the active user is interested in the class under “Rock” though he did not access “Classic” items before.

### 5.1.3 Ratings against Japanese music artists

We also used 58,104 implicit ratings of 2,800 users extracted from blog entries in Doblog using a Japanese taxonomy provided by ListenJapan in the same way as Nakatsuji et al. did for the non-Japanese taxonomy. The Japanese taxonomy contains 153 genres as classes and 7,421 artists as items. The class hierarchy in this taxonomy is as deep as that in the non-Japanese taxonomy. The average number of ratings assigned to each item is 10.8.

## 5.2 Methodology

We randomly divided dataset  $D$  into two datasets: training dataset  $T$  and predicted dataset  $P$ . Thus, we could acquire users who had items whose classes are in  $P$  though they are not included in  $T$ . We then used  $T$  to measure the similarity between users, and created a user graph to measure the relatedness between users on the graph. We

examined four ratios of  $T$  to  $D$ ,  $\frac{T}{D}$ : 0.1, 0.3, 0.5, and 0.7. When  $\frac{T}{D}$  is small, the dataset of user ratings is sparse.

Following the standard evaluation methodology for CF, we predicted the user ratings only on the withheld ratings in  $T$  and computed Mean Absolute Error (MAE), which penalizes each miss by the distance to the actual rating. This measure is written below, where  $n$  is the number of entries in  $P$ , and  $P_i$  and  $R_i$  are the predicted and actual ratings of the  $i$ th entry, respectively.

$$MAE = \frac{\sum_{i=1}^n |P_i - R_i|}{n} \quad (6)$$

We also check the prediction coverage, that is the ratio between the items that are predicted by the method and the items that are included in  $P$ [4]. The prediction coverage can be written as follows:

$$Coverage = \frac{|Items\ in\ P\ found\ by\ the\ algorithm|}{|P|} \quad (7)$$

We investigate the prediction coverage according to the novelty score in detail. Note that the accuracy and item novelty are different metrics and must be measured and/or optimized separately.

## 5.3 Compared methods

We compared our similarity measurement method to the following similarity measures.

- Pearson correlation coefficient (*Pearson*): similarity of users is measured by Pearson correlation coefficient.
- Cosine-based approach (*Cosine*): similarity of users is measured by cosine-based approach.
- Method by Ziegler (*Ziegler*): similarity of users is measured by the method proposed by Ziegler et al.[24, 25]. It measures user similarity only considering co-rating classes and not considering *not* co-rating classes. We set parameter  $\chi$  used in this method properly for each dataset to achieve the most accurate results.
- Taxonomy (Jaccard & Pearson) ( $T(J\&P)$ ): this method is explained in detail in the Method section.

<sup>6</sup><http://www.doblog.com/>; Unfortunately, Doblog terminated services on May 2009.

<sup>7</sup>The music taxonomies in our evaluation can be accessed via ListenJapan home page: <http://listen.jp/>

Table 4: MAE against Japanese music dataset.

$T/D$	0.1			0.3			0.5			0.7		
$\alpha$	Non	0.8	0.6	Non	0.8	0.6	Non	0.8	0.6	Non	0.8	0.6
<i>Pearson</i>	1.29	1.30	1.32	1.19	1.16	1.16	1.20	1.19	1.18	1.18	1.19	1.17
<i>Cosine</i>	1.32	1.34	1.34	1.16	1.17	1.17	1.13	1.15	1.16	1.16	1.13	1.13
<i>Ziegler</i>	1.39	1.38	1.35	1.17	1.17	1.15	1.13	1.13	1.12	1.15	1.15	1.16
$T(J\&P)$	1.29	1.27	1.27	1.11	1.09	1.09	1.12	1.13	1.12	1.15	1.17	1.21
$T(J)$	1.32	1.31	1.31	1.12	1.11	1.09	1.11	1.12	1.12	1.13	1.11	1.10

- Taxonomy (Jaccard) ( $T(J)$ ): the similarity score of ratings between users against classes and those against items are measured using our Jaccard-based method.

## 5.4 Results on accuracy

We first estimated the number of users,  $N$ , similar to the user, and determined the MAE. As a result, we set  $N$  to 40 against the music datasets and  $N$  to 30 against the movie dataset because MAE worsens if  $N$  is large.

Next, we evaluated the accuracy of our method by changing the ratio of training data,  $\frac{T}{D}$  and the restart parameter  $\alpha$ . Results against the movie dataset, those against the non-Japanese music dataset, and those against the Japanese music dataset are shown in Table 2, Table 3, and Table 4, respectively. Here, “Non” in each table indicates the results when RWR was not performed.

For the movie dataset, our method,  $T(J\&P)$  and  $T(J)$ , achieved better accuracy than the other methods when the dataset was sparse, however, method *Pearson* achieved higher accuracy than our methods when the dataset was not sparse. The movie dataset is much more dense than the music dataset, and the taxonomy of movie items is much simpler than that of the music dataset. When we use the music dataset, most of the results of our method,  $T(J\&P)$  and  $T(J)$ , are better than those of the other methods. Thus, we can understand that our methods work better when the dataset is sparse and the taxonomy is detailed. We will enrich the taxonomy of movie dataset by adding additional movie metadata such as actors and directors, and evaluate the movie dataset again in the section 5.6.

Interestingly,  $T(J\&P)$  achieved the highest accuracy against the non-Japanese music dataset. However, when we used the other datasets,  $T(J)$  tends to achieve higher accuracy. This is because the non-Japanese taxonomy is more detailed than the other taxonomies, and this situation well suits our assumption that users who like an item also like classes that include that item. Furthermore, when the dataset is not sparse, users tend to have many items in each class. In such situations, the end class of the taxonomy can be classified into more detailed classes, and  $T(J)$  works better because it checks how many instances shared and not shared by users in each class in the taxonomy. For example,  $T(J)$  considers that the interest of  $a$  and that of a user who likes items,  $I_{21}$ ,  $I_{22}$  and  $I_{23}$ , are not the same. As a result, it can divide similar users who like items  $I_{21}$  and  $I_{22}$ , from users who like items  $I_{21}$ ,  $I_{22}$ , and  $I_{23}$ .

We also investigated the impact of restart parameter  $\alpha$ , and found that most results are improved if we decrease  $\alpha$  to a certain point. In particular, when we used not so dense datasets such as the music datasets, the prediction accuracy is often improved if we decrease  $\alpha$ . When the data is sparse, the user profile is not so well constructed, and thus similarity measurements do not perform so well. In such a case, the

Table 5: Prediction coverage versus item novelty against non-Japanese music dataset.

		Score of novelty						
	$\alpha$	0	1	2	3	4	5	6
<i>Pearson</i>	Non	0.20	0.39	0.39	0.36	0.31	0.28	0.11
	0.6	0.21	0.45	0.41	0.39	0.34	0.29	0.10
<i>Cosine</i>	Non	0.17	0.42	0.56	0.44	0.40	0.53	0.64
	0.6	0.16	0.40	0.54	0.44	0.39	0.53	0.64
<i>Ziegler</i>	Non	0.25	0.46	0.51	0.39	0.29	0.43	0.48
	0.6	0.27	0.51	0.54	0.44	0.32	0.45	0.49
$T(J\&P)$	Non	0.34	0.75	0.75	0.72	0.58	0.67	0.51
	0.6	0.35	0.77	0.79	0.79	0.66	0.78	0.67

Table 6: Prediction coverage versus item novelty against Japanese music dataset.

		Score of novelty						
	$\alpha$	0	1	2	3	4	5	6
<i>Pearson</i>	Non	0.24	0.44	0.31	0.33	0.22	0.10	0.11
	0.6	0.22	0.40	0.31	0.31	0.21	0.10	0.11
<i>Cosine</i>	Non	0.25	0.54	0.66	0.58	0.66	0.37	0.11
	0.6	0.25	0.52	0.66	0.58	0.67	0.37	0.11
<i>Ziegler</i>	Non	0.29	0.62	0.66	0.60	0.61	0.47	0.11
	0.6	0.32	0.68	0.71	0.65	0.64	0.44	0.11
$T(J\&P)$	Non	0.34	0.71	0.76	0.73	0.71	0.49	0.11
	0.6	0.35	0.71	0.75	0.74	0.74	0.52	0.19

accuracy is improved by incorporating the items accessed by users who are related to the active user  $a$  on the graph though the similarity between  $a$  and each of those users is small.

From above results, we consider that our method can identify items accurately when the taxonomy is constructed in detail and the rating dataset is sparser.

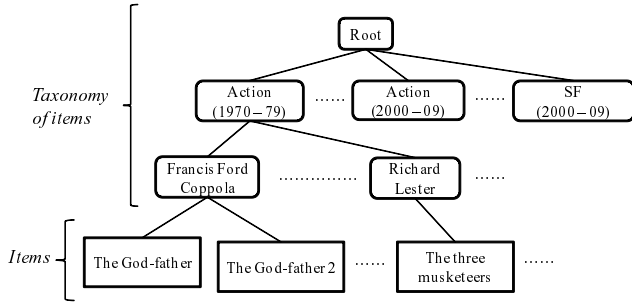
## 5.5 Results on item novelty

We also investigated the prediction coverage of the methods according to the novelty of items and parameter  $\alpha$ . Table 5 and Table 6 show the results for the non-Japanese music dataset with  $\frac{T}{D} = 0.7$  and those for the Japanese music dataset with  $\frac{T}{D} = 0.7$ , respectively.

Those results show that our methods (even without RWR) have better prediction coverage than the other methods because they can measure the similarity of users even if they do not share the same items. However, when RWR is not performed, the results indicate that the items with higher novelty are difficult to identify because users who are very similar to each other tend to share items with low novelty.

We next show what happens when  $\alpha$  is reduced. We found that our taxonomy-based methods offer high prediction cov-





**Figure 3: Example of extended taxonomy of MovieLens items.**

**Table 7: MAE against movie dataset with extended taxonomy.**

$T/D$	0.5			0.7		
$\alpha$	Non	0.8	0.6	Non	0.8	0.6
<i>Pearson</i>	0.95	0.94	0.94	0.96	0.96	0.96
<i>Cosine</i>	0.94	0.93	0.93	0.96	0.95	0.96
<i>Ziegler</i>	0.95	0.95	0.94	0.98	0.98	0.97
$T(J\&P)$	0.92	0.92	0.92	0.95	0.95	0.96
$T(J)$	0.93	0.93	0.93	0.96	0.96	0.96

erage against items with higher novelty. Furthermore, as we demonstrated above, the accuracy does not fall significantly if we decrease parameter  $\alpha$ .

From those results, we consider that our methods can identify items with higher novelty and more accurately than any other methods. Typical recommendations based on previous CF methods present types of items that are similar to those the active user highly rated. Our graph-based method allows the active user to discover items that are located in the class he did not know well through the items accessed by the users who are not directly similar to the active user.

As the reader can naturally imagine, those results also suggest that our method increases the diversity of the recommendation results. We, however, consider that our score of novelty is superior to the notion of diversity. The score of novelty gives the semantic relationships between the present interests of the active user and the item recommended to him. For example, by altering the desired minimum novelty score, the user can winnow the recommendation results to increase user satisfaction.

## 5.6 Results when extending movie taxonomy

We extended the MovieLens item taxonomy by matching items in the MovieLens dataset with those provided by one of the Japanese movie information provider, goo movie. This was done by checking the title name of items. If the name of a MovieLens item matched the name of an item provided in goo movie, we extended the item taxonomy by creating classes that have the properties of genres attached by MovieLens and those of metadata attached by goo movie such as release date, actors, directors, writer, and musician. Fig. 3 presents an example. The item “The Godfather” is originally classified in “Action” class in the MovieLens dataset. We extend it by using the metadata provided by goo movie, and create a new class “Action (1970-79)” and we also create a class “Francis Ford Coppola” under the class “Action (1970-

**Table 8: MAE against restaurant dataset.**

$T/D$	0.5			0.7		
$\alpha$	Non	0.8	0.6	Non	0.8	0.6
<i>Pearson</i>	0.65	0.55	0.54	0.54	0.53	0.49
<i>Cosine</i>	0.68	0.65	0.64	0.59	0.59	0.58
<i>Ziegler</i>	0.52	0.53	0.54	0.47	0.46	0.46
$T(J\&P)$	0.49	0.49	0.50	0.45	0.45	0.44

79)”. Then, we classified “The Godfather” to “Francis Ford Coppola”. This yields automatic enrichment of the taxonomy in the MovieLens Dataset. As a result, we acquired a taxonomy with 8,889 classes and 922 items. The number of items is less than the original dataset because some items had no match in the goo movie item name set. The number of users, 943, was the same as the original MovieLens dataset.

Next, we evaluated the accuracy of our method by using the MovieLens dataset with extended taxonomy. Results are shown in Table 7. The extension allowed our method to achieve higher accuracy than all compared methods. Thus, we conclude that deeper taxonomies are good for predicting user interests. Note that the results are not so good as the results using the original taxonomy shown in Table 2. This is because there are fewer items to be learned which decreases accuracy.

## 5.7 Results when using restaurant dataset

For completeness, we evaluated our method by applying it to restaurant ratings.

We used a dataset from Tabelog to evaluate the performance of the proposed method. We focused on restaurants in Tokyo among all restaurants known to Tabelog. The tested dataset contained 401,584 ratings of 7,403 anonymous Tabelog users on 133,432 restaurants. The average number of ratings assigned to each item was 4.64. The taxonomy of Tabelog is quite deep; it has 318 genres as classes, and has three or four hierarchy levels. For example, the end classes of this taxonomy have genres such as “Wine bar” and “Beer garden”. The maximum user rating was 5 and the minimum rating was 0. Like the MovieLens dataset, a low item rating indicates a negative interest in that item.

The taxonomy of Tabelog is quite deep, so we apply our method of  $T(J\&P)$  to this dataset according to the results discussed in section 5.4. We also apply our method to this dataset while changing the ratio of  $T$  to  $D$ ,  $\frac{T}{D} = 0.5$  and 0.7, i.e. the dataset is not sparse.

We first estimated the number of users,  $N$ , similar to the user, and determined the MAE. As a result, we set  $N$  to 40 against this dataset.

Results against MAE are shown in Table 8. Our method achieves the highest accuracy among all compared methods. We also investigated the prediction coverage of the methods according to the novelty of items and parameter  $\alpha$ . Table 9 shows the results with  $\frac{T}{D} = 0.7$ . They show that our methods have better prediction coverage than the other methods even if RWR is not used. Furthermore, we found that our taxonomy-based methods offer high prediction coverage against items with higher novelty when parameter  $\alpha$  is low.

It is clear from the above that our method can be well applied to item domains other than movies and music.

**Table 9: Prediction coverage versus item novelty against restaurant dataset.**

		Score of novelty						
	$\alpha$	0	1	2	3	4	5	6
Pearson	Non	0.03	0.16	0.21	0.19	0.20	0.22	0.20
	0.6	0.03	0.24	0.24	0.24	0.25	0.26	0.23
Cosine	Non	0.03	0.15	0.19	0.17	0.18	0.20	0.20
	0.6	0.03	0.16	0.20	0.18	0.20	0.21	0.23
Ziegler	Non	0.03	0.15	0.14	0.12	0.10	0.07	0.05
	0.6	0.04	0.23	0.21	0.19	0.17	0.13	0.10
$T(J\&P)$	Non	0.05	0.33	0.34	0.33	0.32	0.31	0.26
	0.6	0.05	0.32	0.34	0.32	0.33	0.33	0.33

## 6. CONCLUSION

This paper introduced a method that can identify, for the active user, items with high novelty. To measure the similarity of users, our method uses items rated by users as well as the taxonomy of items. It then builds a user graph whose nodes are users, and sets weighted edges between user nodes according to the similarity of users. It performs Random Walk with Restarts over the graph, and extracts users who are *not similar* to the active user but whose nodes are arrived at frequently. By incorporating the items accessed by those extracted users, our method identifies items with high novelty for the active user. Evaluations show that our method can identify items with higher novelty and more accurately than existing methods. The time spent in performing RWR on the graph is one problem in identifying items with high novelty, however a fast RWR computation technique was proposed by Tong et al.[21]. The definition of novelty given here may be too rigid and better performance might be achieved by allowing items in classes the user has accessed before (but very infrequently) to be recommended to the user as being novel.

## 7. REFERENCES

- [1] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. UAI'98*, pages 43–52, 1998.
- [2] F. Fouss, A. Pirotte, J. michel Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19:355–369, 2006.
- [3] M. Gori and A. Pucci. Itemrank: A random-walk based scoring algorithm for recommender engines. In *Proc. IJCAI'07*, pages 2766–2771, 2007.
- [4] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.
- [5] A. Kohrs and B. Meriardo. Clustering for collaborative filtering applications. In *Computational Intelligence for Modelling, Control & Automation*. IOS. Press, 1999.
- [6] I. Konstantas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *Proc. SIGIR'09*, pages 195–202, 2009.
- [7] Y. Koren. Collaborative filtering with temporal dynamics. In *Proc. KDD'09*, pages 447–456, 2009.
- [8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [9] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proc. ICML'09*, pages 617–624, 2009.
- [10] L. Lovasz. *Random Walks on Graphs: A Survey*, volume 2. Bolyai Society for Mathematical Studies, 1996.
- [11] C. D. Manning and H. Schtze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1 edition, 1999.
- [12] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *Proc. CHI'06*, pages 1097–1101, 2006.
- [13] M. Nakatsuji, Y. Fujiwara, A. Tanaka, T. Uchiyama, and T. Ishida. Recommendations over domain specific user graphs. In *Proc. ECAI'10*, 2010.
- [14] M. Nakatsuji, M. Yoshida, and T. Ishida. Detecting innovative topics based on user-interest ontology. *Web Semantics*, 7(2):107–120, 2009.
- [15] K. Onuma, H. Tong, and C. Faloutsos. Tangent: a novel, 'surprise me', recommendation algorithm. In *Proc. KDD'09*, pages 657–666, 2009.
- [16] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proc. IUI'02*, pages 127–134, 2002.
- [17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. CSCW'94*, pages 175–186, 1994.
- [18] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proc. WWW'01*, pages 285–295, 2001.
- [19] V. Schickel-Zuber and B. Faltings. Inferring user's preferences using ontologies. In *Proc. AAAI'06*, pages 1413–1418, 2006.
- [20] V. Schickel-Zuber and B. Faltings. Oss: A semantic similarity function based on hierarchical onotologies. In *Proc. IJCAI'07*, pages 551–556, 2007.
- [21] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *Proc. ICDM'06*, pages 613–622, 2006.
- [22] L. Ungar and D. Foster. Clustering methods for collaborative filtering. In *Proceedings of the Workshop on Recommendation Systems*. AAAI Press, Menlo Park California, 1998.
- [23] H. Yildirim and M. S. Krishnamoorthy. A random walk method for alleviating the sparsity problem in collaborative filtering. In *Proc. RecSys'08*, pages 131–138, 2008.
- [24] C. N. Ziegler, G. Lausen, and L. S. Thieme. Taxonomy-driven computation of product recommendations. In *Proc. CIKM'04*, pages 406–415, 2004.
- [25] C. N. Ziegler and S. M. McNee. Improving recommendation lists through topic diversification. In *Proc. WWW'05*, pages 22–32, 2005.