Ontology-Based Collaborative Recommendation

Ahu Sieg, Bamshad Mobasher, Robin Burke

Center for Web Intelligence
DePaul University, Chicago, Illinois, USA
{asieg, mobasher, rburke}@cdm.depaul.edu

Abstract. Recommender systems have emerged as critical tools that help alleviate the burden of information overload for users. Since these systems have to deal with a variety of modes of user interactions, collaborative recommendation must be sensitive to a user's specific context and changing interests over time. Our approach to building context-sensitive collaborative recommendation is a hybrid one that incorporates semantic knowledge in the form of a domain ontology. User profiles are defined relative to the ontology, giving rise to an *ontological user profile*. In this paper, we describe how ontological user profiles are learned, incrementally updated, and used for collaborative recommendation. We empirically show that the ontological approach significantly improves the accuracy and coverage of recommendations.

1 Introduction

Recommender systems [1] have become essential tools in assisting users to find what they want in increasingly complex information spaces. Collaborative recommender systems typically generate recommendations by identifying neighborhoods for the target user consisting of other users with similar interests or preferences [2].

Typical collaborative recommenders rely on profiles of users represented as flat vectors of ratings or preference scores. Thus, the same collection of user preferences across all items or resources is used as the basis for generating recommendations regardless of the user's current information context or task-related needs. Consider the following example. Suppose Steve buys and rates mystery-detective fiction novels for his own entertainment ("Da Vinci Code"), books on computer science topics ("Python Programming") for work-related purposes, children's books ("Green Eggs and Ham") for his daughter. It makes little sense to represent Steve's *interest in books* in a single representation that aggregates all of these disparate interests without some acknowledgment that they represent different sorts of needs and contexts. The system needs to know the difference between computer books and children's books, as well as Steve's current context (buying a book for his personal reading or for his work), in order to make the most useful recommendation. Furthermore, a system that is aware of this difference may also have the capability of recognizing similarities among syntactically disparate items, and be able to recommend a book on Perl scripting to Steve because he has shown an interest in Python programming.

This scenario exemplifies why it is desirable for intelligent and personalized information systems to be capable of seamlessly integrating knowledge from three sources: the short-term user activity, representing immediate user interests; long-term user profiles, representing established preferences; and existing ontologies that provide an explicit representation of the domain of interest. Such systems will be able to leverage a variety of sources of evidence to provide the best personalized experience for the user, including both the semantic evidence associated with the user's individual interaction, as well as social knowledge derived collaboratively from peer users.

In this paper, we present an approach to collaborative recommendation that effectively incorporates semantic knowledge from ontologies with collaborative user preference information. The salient feature of our framework is the notion of *ontological user profiles* which are instances of a pre-existing domain ontology with numerical annotations associated with concepts derived from users' past behavior and preferences. The ontology represents concepts and relationships in a particular domain of interest, books for example. In this paper, we use the term ontology to refer to a hierarchical concept structure and instances within the knowledge base. Rather than being associated with single atomic entities like individual books, users' choices and preferences are associated with relevant concepts in the ontology. So, the fact that Steve buys computer books such as "Python Programming" can be readily distinguished from his interest in children's books because they occupy disparate places in the book ontology.

We present an algorithm based on spreading activation to incrementally update these user profiles, as a result of ongoing user interaction, in a way that takes into account relationships among concepts in the ontology as well as the collaborative evidence derived from the ontological profiles of similar users. Our approach to recommendation generation is an extension of standard user-based collaborative framework in which user similarities are computed based on their interest scores across ontology concepts, instead of their ratings on individual items. Our experimental results for collaborative recommendation, based on real ratings in the book domain, show significant improvement in prediction accuracy as well as coverage when compared to standard collaborative filtering.

2 Related Work

Widely used collaborative filtering methods can be divided into two main categories including Memory-based (user-based) and Model-based (item-based) algorithms [3, 4]. User-based techniques [5] generally model the user as a vector of item ratings and compare these vectors using a correlation or similarity measurement. Item-based algorithms [6] explore the relationships among items first, rather than the relationships between users, thus avoiding the bottleneck of having to search for neighbors among a large user population of potential neighbors.

Content-based filtering methods [7] have also been used in the context of recommending books and Web pages, where content descriptors are available. Rather than using simple feature vector models, our work differs from existing approaches by taking advantage of the deeper semantic knowledge in an existing ontology for generating recommendations.

Many recommender systems suffer from the cold-start problem of handling new items or new users. Hybrid recommenders [8] combine semantic or content-knowledge with collaborative filtering to deal with this problem. Knowledge-based recommender systems use knowledge about users and products to pursue a knowledge-based approach to generating a recommendation, reasoning about what products meet the user's requirements [9]. Our work can be described as a knowledge-based collaborative hybrid.

The availability of large product taxonomies such as *Amazon.com* and *Open Directory Project* has allowed researchers to incorporate semantic information into recommender systems [10]. In order to address rating sparsity, Ziegler et al. [11] classify products by topics based on taxonomic information. Cho and Kim [12] have utilized a product taxonomy to overcome scalability issues. In [13], spreading activation techniques are used to find related concepts in the ontology given an initial set of concepts and corresponding initial activation values.

In our approach, the hierarchical structure of an underlying ontology is used explicitly and automatically in the learning and incremental updating of user profiles. There has been little work in the area of ontological user modeling and even less in the application of such models to Web personalization [14]. Our research follows the lead of other systems [15] that use ontologies to mediate information access, but these systems have generally not incorporated user modeling.

3 Augmenting Collaborative Recommendation

We take the goal of the recommender system to be the presentation of personalized recommendations for a particular target user. To accomplish this task, there are three broad categories of knowledge that may come into play: social, individual, and content knowledge [16]. Social knowledge covers what we know about the large community of users other than the target user, whereas individual knowledge refers to what we know about the target user. Content knowledge encapsulates domain knowledge about the items being recommended.

Recommender systems based on collaborative filtering utilize explicit or implicit ratings collected from a population of users. The *standard k-Nearest Neighbor (kNN)* algorithm operates by selecting the k most similar users to the target user, and formulates a prediction by combining the preferences of these users. Without the advantage of deeper domain knowledge, collaborative filtering models are limited in their ability to reason about the relationships between item features and about the underlying factors contributing to the final recommendations.

Our goal is to augment collaborative filtering by incorporating domain knowledge in the form of an ontology to enhance personalized recommendations. The ability to learn from user interaction is a critical factor for a good recommender system. In our ontology-based user model, the user behavior is represented not as entries in a uniform vector, but as annotations to an ontology. We refer to this structure as the ontological user profile. In our previous work [17], ontological user profiles are utilized for Web search personalization based on individual users' interests. In this paper, we focus on a collaborative approach for ontology-based recommendation.

We maintain and update the ontological user profiles based on the user behavior and on-going interaction. For example, when Steve buys a book on programming in Python, the user profile associates this fact with the Python programming language concept via an annotation, and may activate other nearby concepts such as the Perl programming language. The system would not, for example, activate nodes associated with snakes or with British comedy troupes, although these have a syntactic relationship to the word "python". We utilize profile normalization so that the relative importance of concepts in the profile reflect the changing interests and varied information contexts of the user.

An ontological approach to user profiling has proven to be successful in addressing the *cold-start problem* in recommender systems where no initial information is available early on upon which to base recommendations [18]. Using ontologies as the basis of the profile allows the initial user behavior to be matched with existing concepts in the domain ontology and relationships between these concepts. Therefore, our approach strengthens the knowledge sources discussed above by providing an enriched representation of social and individual knowledge. Rather than developing the domain ontology ourselves, we rely on existing hierarchical taxonomies such as *Amazon.com's Book Taxonomy*.

Since collaborative filtering is based on the ratings of the neighbors who have similar preferences, it is very important to select the neighbors properly to improve the quality of the recommendations. Rather than computing user similarity on the whole set of items, we use a completely novel approach where the similarity among users is computed based on the users' level of interest for each concept. We compare the ontological user profiles for each user to form semantic neighborhoods. Because the number of items is often very large and so is the diversity among items, users who have similar preferences in one category may have totally different judgments on items of another kind [19]. Our approach allows us to take advantage of the deeper semantic knowledge in the domain ontology when selecting neighbors based on the interest level for each concept in the user profiles.

4 Ontology-Based Personalized Recommendation

For our purposes, an ontology is simply a hierarchy of topics, where the topics can be used to classify items being recommended. There is one main ontology on which all user profiles are based – we call this the *reference ontology*. An ontological user profile is a set of nodes from the reference ontology, each annotated with an *interest score*, which represent the degree of interest that the user has expressed in that topic or concept. Each node in the ontological user profile is a pair, $\langle C_j, IS(C_j)\rangle$, where C_j is a concept in the ontology and $IS(C_j)$ is the interest score annotation for that concept. Whenever the system acquires new evidence about user interests, such as purchases, page views, or explicit ratings, the user profile is updated with new interest scores.

The hierarchical relationship among the concepts is taken into consideration for maintaining the ontological user profiles as we update the annotations for existing concepts. Each concept in the user profile is annotated with an *interest score* which has an initial value of one. As the user interacts with the system (i.e. rating a new book), the ontological user profile is updated and the annotations for existing concepts are modified. As a result, the profiles are maintained and updated incrementally based on the user's ongoing behavior.

4.1 Learning Profiles by Spreading Activation

We use *Spreading Activation* to incrementally update the *interest score* of the concepts in the user profiles. In our current implementation, the users' item based ratings are utilized to propagate interest scores in the user profiles. The process of learning an ontological user profile is depicted in Figure 1 using a portion of the ontology as an example.

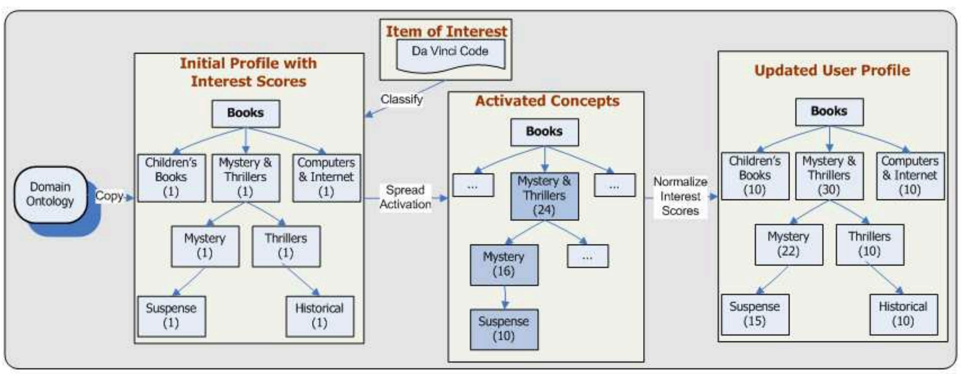


Fig. 1. Updating an Ontological User Profile

We use a very specific configuration of spreading activation, depicted in Algorithm 1, for the sole purpose of maintaining *interest scores* within a user profile. The ontological user profile is treated as the semantic network and the interest scores are updated based on activation values. The algorithm has an initial set of concepts from the ontological user profile. The main idea is to activate other concepts following a set of weighted relations during propagation and at the end obtain a set of concepts and their respective activations.

As any given concept propagates its activation to its neighbors, the weight of the relation between the origin concept and the destination concept plays an important role in the amount of activation that is passed through the network.

Algorithm 1: Spreading Activation Algorithm

```
Input: Ontological user profile with interest scores and an item of interest to the user, i
Output: Ontological user profile with updated interest scores
CON = {C_1, ..., C_n}, user profile concepts with interest scores
IS(C_j) and Activation(C_j), interest score and activation value for concept C_j
// Step 1: Spreading Activation
Initialize priorityQueue;
Set initial Activation of all concepts to 0;
foreach C_j ∈ CON do
    begin
        if (i ∈ C_j) then
            Activation(C_j) = IS(C_j);
            priorityQueue.Add(C_j);
        end
    end
end
while priorityQueue.Count > 0 do
    Sort priorityQueue; // activation values(descending)
    C_j = priorityQueue[0]; // first item(spreading concept)
    priorityQueue.Dequeue(C_j); // remove item
    if passRestrictions(C_j) then
        linkedConcepts = GetLinkedConcepts(C_j);
        foreach C_l in linkedConcepts do
            Activation(C_l)+ = Activation(C_j) * Weight(C_j, C_l);
            priorityQueue.Add(C_l);
        end
    end
end
// Step 2: Profile Normalization
foreach C_j ∈ CON do
    IS(C_j) = IS(C_j) + Activation(C_j);
    n = sqrt(n + (IS(C_j))^2); // square root of sum of squared interest scores
end
foreach C_j ∈ CON do
    IS(C_j) = (IS(C_j) * k)/n; // normalize to constant length, k
end
```

Thus, a one-time computation of the weights for the relations in the network is needed. Since the nodes are organized into a concept hierarchy derived from the domain ontology, we compute the weights for the relations between each concept and all of its subconcepts using a measure of containment. The weight, $Weight(C_j, C_s)$, of the relation for concept C_j and one of its subconcepts C_s is computed based on the number of items that are categorized under each concept. Once the weights are computed, we normalize the weights to ensure that the total sum of the weights of the relations between a concept and all of its subconcepts equals to one.

The algorithm is executed for each item of interest, such as a book. For each iteration of the algorithm, the initial activation value for each concept in the user profile is reset to zero. The concepts which contain the specific item are activated and the activation value, $Activation(C_j)$, for each activated concept C_j is set to the existing interest score, $IS(C_j)$, for that specific concept. If there is no interest information available for a given concept, then $IS(C_j)$ equals to one. The concept with the highest activation value gets removed from the queue after propagating its activation to its neighbors. The amount of activation that is propagated to each neighbor is proportional to the weight of the relation. The neighboring concepts which are activated and are not currently in the priority queue are added to queue, which is then reordered. The process repeats itself until there are no further concepts to be processed. For a given spreading concept, we can ensure the algorithm processes each edge only once by iterating over the linked concepts only one time. The order of the iteration over the linked concepts does not affect the results of activation. The linked concepts that are activated are added to the existing priority queue, which is then sorted with respect to activation values.

After spreading activation, the interest scores in the profile are normalized. First the resulting activation values are added to the existing interest scores. The interest scores for all concepts are then treated as a vector, which is normalized to a unit length using a pre-defined constant, k, as the length of the vector. The effect of normalization is to prevent the interest scores from continuously escalating throughout the network. As the user expresses interests in one set of concepts, the scores for other concepts may decrease. For the long-term maintenance, the concepts in the ontological user profile are updated with the normalized interest scores.

4.2 Semantic Neighborhoods and Prediction Computation

In standard collaborative filtering, the similarity between the target user, u, and a neighbor, v, is calculated by the Pearson's correlation coefficient. Our alternative similarity metric uses the interest scores of these users' corresponding ontological profiles. First, we turn the ontological user profiles into flat vectors of interest scores over the space of concepts. We then compare the user profiles to figure out how distant each user's profile is from all other users' profiles. The distance between the target user, u, and a neighbor, v, is calculated by the Euclidean distance formula defined below:

$$distance_{u,v} = \sqrt{\sum_{j \in C} (IS(C_{j,u}) - IS(C_{j,v}))^2}$$

where C is the set of all concepts in the reference ontology, $IS(C_{j,u})$ and $IS(C_{j,v})$ are the interest scores for concept C_j for the target user u and neighbor v, respectively. Once all distances have been computed, we normalize the distance between the target user u and a neighbor v, then calculate a similarity value based on the inverse of the normalized distance.

The most similar k users are selected to generate the semantic neighborhoods. To further improve the quality of the neighborhoods, we use a concept-based filtering for the neighbors where a neighbor is included in the final prediction algorithm only if that neighbor's interest score for the specific concept is greater than their mean interest scores in their user profile. Our resulting semantic neighborhoods are not only based on similar users' explicit ratings for an item, but also based on the degree of interest those users have shown for the topic of a given item.

The ability to generate good recommendations relies heavily on the accurate prediction of a user's rating for an item they have not seen before. Our prediction algorithm uses a variation of Resnick's standard prediction formula [4] defined below:

$$p_{u,i} = \bar{r}_u + \frac{\sum\limits_{v \in V} sim_{u,v} * (r_{v,i} - \bar{r}_v)}{\sum\limits_{v \in V} sim_{u,v}},$$

where $\bar{r}_u$ is the mean rating for the target user, V is the set of k similar users, $\bar{r}_v$ is the mean rating for a neighbor, $sim_{u,v}$ is the similarity described above.

We utilize the semantic evidence in the ontology for computing the mean rating for a user. For the mean rating of a target user or one of its neighbors, $\bar{r}_u$ and $\bar{r}_v$ respectively, we maintain two different values including the user's overall mean rating and user's concept-based mean rating. If an item belongs to only one concept, the user's concept-based mean rating is the user's average rating for all books that belong to that specific concept. In the case where a book belongs to multiple concepts, the concept-based mean rating becomes the user's average rating for all books that belong to these concepts. If the user's concept-based mean rating does not exist, the prediction formula uses the user's overall mean rating. Otherwise, the user's concept-based mean rating is used.

5 Experimental Evaluation

In the research community, the performance of a recommender system is mainly measured based on its accuracy with respect to predicting whether a user will like a certain item or not [20]. Our experimental evaluation focuses on comparing the quality of the recommendations based on our ontological approach versus standard collaborative filtering.

5.1 Experimental Data Sets and Metrics

Our data set consists of a subset of the book ratings that were collected by Ziegler in a 4-week crawl from the Book-Crossing community[11]. For each distinct ISBN, a unique identifier for the books in the dataset, we mined *Amazon.com's Book Taxonomy* and collected the category, title, URL, and editorial reviews for the specific book. Our resulting reference ontology includes 4,093 concepts and a total of 75,646 distinct books that are categorized under various concepts.

Only the explicit ratings, expressed on a scale from 1-10, are taken into account in our experiments. Our data set includes 72,582 book ratings belonging to those users with 20 or more ratings. The data set was converted into a user-item matrix that had 1,110 rows (i.e. users) and 27,489 columns (i.e. books). For evaluation purposes, we used 5-Fold cross-validation. For each fold, 80% of the book ratings were included in the *training set*, which was utilized to compute similarity among users. The remaining 20% of the book ratings were included in the *test set*, which was used for predicting ratings. The advantage of *K-Fold cross-validation* is that all the examples in the dataset are eventually used for both training and testing.

To measure prediction accuracy, we rely on a commonly used metric *Mean Absolute Error (MAE)*, which measures the average absolute deviation between a predicted rating and the user's true rating: $MAE = \frac{\sum |p_{u,i} - r_{u,i}|}{N}$, where N is the total number of ratings over all users, $p_{u,i}$ is the predicted rating for user u on item i, and $r_{u,i}$ is the actual rating. The lower the *MAE*, the more accurately a recommender systems predicts user ratings. One main advantage of *MAE* is that it is a statistical metric which allows for testing the significance of a difference between the mean absolute errors of two systems. For our second type of evaluation, we generate a list of *Top-N recommendations* for each user. We compare the recommendations to the user's actual preferences and consider each match a hit. We use a *Hit Ratio* metric to compare our approach to standard collaborative filtering.

5.2 Experimental Methodology and Results

The first step in our experimental evaluation was to compute user-to-user similarity for the *standard kNN* algorithm using the Pearson's correlation based on the training data. Next, we used the books in the training set to generate ontological user profiles. Each user started out with an ontological user profile where all interest scores were initialized to one, this simulates a situation where no initial user interest information is available. For each book that was rated by a user, we performed our spreading activation algorithm to update interest scores in the ontological user profile for that specific user. In order to ensure the interest in the profiles is propagated based on strong positive evidence, only the items with ratings that were equal to or greater than the user's overall average rating were utilized for spreading activation. After an ontological user profile was created for each user based on their ratings, we utilized our semantic neighborhood generation approach explained above to compute the similarity among user profiles.

We calculated the *MAE* across the predicted ratings produced by each algorithm. For both the *standard kNN* and our ontological approach, the most similar k users were selected to compute the prediction for each item in the test set.

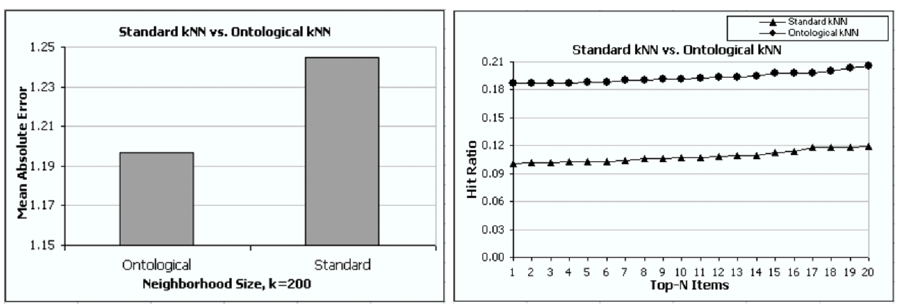
To generate a recommendation list for a specific user, we computed a predicted rating for all items that were rated by that user's neighbors, excluding the items that were rated by the user. With this type of an evaluation, the goal is to generate recommendations that the user has not seen before. The recommendation list was sorted in descending order with respect to the predicted rating for each item. Therefore, items with higher predicted ratings are included in the *Top-N recommendations*. We compared the recommendation list to the user's actual ratings for items in the test set.

We ran our experiments for different values for the neighborhood size k, ranging from 20 to 200. For each value of k, the *MAE* was lower for predictions using our ontological approach than the *MAE* across the predictions generated with the *standard kNN* algorithm. Our ontological approach also provides much better coverage, which is a measure for the percentage of items that a recommender system can provide predictions for. As depicted in Table 1, we computed three

Algorithm	Overall Ratings	Actual Ratings	Default Ratings	Coverage
Standard kNN	1.139	1.245	1.049	45.9%
Ontological kNN	1.112	1.197	1.025	50.6%

Table 1. Mean Absolute Error, k = 200 - Standard kNN vs. Ontological Approach

different *MAE* values for each algorithm using overall ratings, actual ratings, and default ratings. The *MAE* across actual ratings takes into account only those ratings where an actual predicted rating can be made based on the ratings of neighbors as opposed to the predicted ratings based on the user's default rating due to lack of ratings from neighbors.



(a) Mean Absolute Error

(b) Hit Ratio

Fig. 2. Standard kNN vs. Ontological Approach, k = 200

Inspecting the *MAE* across the actual predicted ratings separately than the default ratings is important in order to effectively compare the different algorithms presented in this paper. In [21], the authors explore the importance of the influence of neighbors in collaborative filtering and present their finding on three commonly used, large-scale, real-world datasets including MovieLens, NetFlix, and BookCrossing. Due to the high sparsity of the Book-Crossing dataset, user-based collaborative filtering performs particularly poorly, with only 53% of neighborhood estimates actually contributing to better quality predictions than chance [21]. One additional advantage of our approach is that we are able to improve the predicted ratings based on the user's default rating since we use a concept-based mean as opposed to taking the user's average across all of the items rated by that user.

The comparative results with the *MAE* values across actual predicted ratings for $k = 200$ are depicted in Figure 2(a). The *MAE* values were confirmed to be significantly different using the *ANOVA* significance test with a 99% confidence interval, *p-Value = 6.9E-11*. Thus, we can confidently conclude that the prediction accuracy using our ontological approach is higher than the prediction accuracy of the *standard kNN* algorithm.

Next, we present our *Hit Ratio* results to compare *standard kNN* with our ontological approach in terms of *Top-N Recommendation*. The *Hit Ratio* is computed by determining whether a hit exists within the top N items in the list for each value of N, where $N = 1$ through $N = 20$. With this approach, the *Hit Ratio* is either 0 or 1 for each value of N for each user. We then take an average across all users in our data set. The recommendation lists for each user were sorted in descending order with respect to the predicted rating for each item. For each algorithm, the *Hit Ratio* results were based on the predicted ratings using a neighborhood size of $k = 200$. As depicted in Figure 2(b), the *Hit Ratio* for *ontological kNN* is significantly improved over *standard kNN*. These results further validate that our ontological approach performs better as a recommender system.

6 Conclusions and Outlook

We have presented our approach to collaborative recommendation that effectively incorporates semantic knowledge from ontologies with collaborative user preference information. Our approach not only outperforms traditional collaborative filtering in prediction accuracy but also offers improvements in coverage. Although accuracy metrics are important, in order to fully satisfy a user's recommendation needs, other measures such as diversity of recommendation lists and uniqueness of recommended items must be considered. In our future work, we plan to further evaluate the advantages of our ontological approach in terms of coverage, diversity, personalization, and cold-start performance.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions. on Knowledge and Data Engineering **17**(6) (2005) 734–749
2. Schafer, J.B., Frankowski, D., Herlocker, J.L., Sen, S.: Collaborative filtering recommender systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007)
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, San Francisco, CA (1998) 43–52
4. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work, CSCW 1994, New York, NY (2004) 175–186

5. Herlocker, J.L., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval, SIGIR 1999, Berkeley, CA (August 1999)
6. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, WWW '01, New York, NY (2001) 285–295
7. Mooney, R., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the fifth ACM conference on Digital libraries, DL 2000, New York, NY (2000) 195–204
8. Burke, R.: Hybrid web recommender systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007)
9. Burke, R.: Knowledge-based recommender systems. In Kent, A., ed.: Encyclopedia of Library and Information Systems. Volume 69. Marcel Dekker (2000)
10. Middleton, S., Shadbolt, N., Roure, D.C.D.: Ontological user profiling in recommender systems. ACM Transactions on Information Systems **22**(1) (2004) 54–88
11. Ziegler, C., McNee, S., Konstan, J., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan (May 2005)
12. Cho, Y., Kim, J.: Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. Expert Systems with Applications **26**(2) (2004) 233–246
13. Rocha, C., Schwabe, D., Aragao, M.P.: A hybrid approach for searching in the semantic web. In: Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA (2004) 374–383
14. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In: The adaptive web, LNCS 4321. (2007) 54–89
15. Gauch, S., Chaffee, J., Pretschner, A.: Ontology-based personalized search and browsing. Web Intelligence and Agent Systems **1**(3-4) (2003)
16. Burke, R., Ramezani, M.: Matching recommendation domains and technologies. In Kantor, P., Ricci, F., Rokach, L., Shapira, B., eds.: Handbook of Recommender Systems, to appear. Springer (2010)
17. Sieg, A., Mobasher, B., Burke, R.: Web search personalization with ontological user profiles. In: ACM Sixteenth Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal (November 2007)
18. Middleton, S., Shadbolt, N., Roure, D.D.: Capturing interest through inference and visualization: Ontological user profiling in recommender systems. In: Proceedings of the International Conference on Knowledge Capture, K-CAP 2003, Sanibel Island, Florida (October 2003) 62–69
19. Truong, K., Ishikawa, F., Honiden, S.: Improving accuracy of recommender system by item clustering. IEICE - Transactions on Information and Systems **E90-D**(9) (2007) 1363–1373
20. Herlocker, J.L., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems **22**(1) (2004) 5–53
21. Rafter, R., O'Mahony, M., Hurley, N.J., Smyth, B.: What have the neighbours ever done for us? a collaborative filtering perspective. In: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization, UMAP 2009, Trento, Italy (2009) 355–360