Semantically Enhanced Collaborative Filtering on the Web

Bamshad Mobasher, Xin Jin, and Yanzan Zhou
{mobasher,xjin,yzhou}@cs.depaul.edu

Center for Web Intelligence
School of Computer Science, Telecommunication, and Information Systems
DePaul University, Chicago, Illinois, USA

Abstract. Item-based Collaborative Filtering (CF) algorithms have been designed to deal with the scalability problems associated with traditional user-based CF approaches without sacrificing recommendation or prediction accuracy. Item-based algorithms avoid the bottleneck in computing user-user correlations by first considering the relationships among items and performing similarity computations in a reduced space. Because the computation of item similarities is independent of the methods used for generating predictions, multiple knowledge sources, including structured semantic information about items, can be brought to bear in determining similarities among items. The integration of semantic similarities for items with rating- or usage-based similarities allows the system to make inferences based on the underlying reasons for which a user may or may not be interested in a particular item. Furthermore, in cases where little or no rating (or usage) information is available (such as in the case of newly added items, or in very sparse data sets), the system can still use the semantic similarities to provide reasonable recommendations for users. In this paper, we introduce an approach for semantically enhanced collaborative filtering in which structured semantic knowledge about items, extracted automatically from the Web based on domain-specific reference ontologies, is used in conjunction with user-item mappings to create a combined similarity measure and generate predictions. Our experimental results demonstrate that the integrated approach yields significant advantages both in terms of improving accuracy, as well as in dealing with very sparse data sets or new items.

1 Introduction

The continued growth and increasing complexity of Web-based applications, from e-commerce, to Web services, to dynamic content providers; has led to a proliferation of personalization tools on a variety of sites. Personalized services, such as recommender systems, help engage visitors, turn casual browsers into customer, or help visitor to more effectively locate pertinent information. Collaborative filtering (CF) [25, 14, 5, 11] is one of the most successful and widely used technologies in personalization and recommender systems.

Traditionally, CF-based systems compare a representation of an active user's preferences (such as explicit ratings on items or implicit navigational patterns) with the historical records of past users to find the k most similar *neighbors* of the active user. These historical records are then used to predict the preference value of the active user on a particular, yet to be rated or visited, item; or to recommend the top N items in which the user may be interested. Since the focus of such systems is on comparing the correlations or similarities among users, they are often referred to as *user-based collaborative filtering* systems.

Despite their success and popularity, traditional CF-based techniques suffer from some well-known limitations [24]. One of the critical limitations is the lack of scalability of the underlying memory-based k-nearest-neighbor approach which requires that the neighborhood formation phase be performed as an online process. For very large data sets this may lead to unacceptable latency for providing recommendations. The scalability problems are further accentuated when collaborative filtering is used in the context of Web usage data. In this case, users' browsing patterns are used to implicitly obtain measures of content preference. For frequent visitors the size of user or session vectors tends to be much larger than in the case of e-commerce purchase patterns. Performing user-user similarity computations in this context further degrades the system performance.

Another important limitation of CF-based systems emanates from the sparse nature of the underlying datasets. As the number of items in the database increases, the density of each user record with respect to these items will decrease. This, in turn, will decrease the likelihood of a significant overlap of visited or rated items among pairs of users, resulting in less reliable computation of correlations, and thus less reliable predictions.

Finally, a significant shortcoming of such systems is their inability to provide recommendations or predictions for new or recently added items: a user's rating on a new item cannot be compared with the ratings of other users on the same item. Furthermore, the system can never generate predictions for new items which have not yet been visited or rated by (a sufficient number of) other users. This problem is often referred to as the "new item problem".

A number of optimization strategies have been proposed and employed to remedy the scalability and sparsity problems associated with collaborative filtering. These strategies include similarity indexing [1] to reduce real-time search costs, and dimensionality reduction methods based on Latent Semantic Indexing (LSI) to alleviate the data sparsity in the user-item mappings [24, 22]. Other approaches have focused on model-based techniques which use machine learning techniques, such as unsupervised clustering of user records [19] or supervised classification models [5]. These approaches separate the offline tasks of creating user models from the real-time task of recommendation generation, thus improving scalability. However, this is sometimes at the cost of lower recommendation accuracy.

In the context of click-stream and e-commerce data, Web usage mining [26] techniques, such as clustering and association rule discovery, that rely on offline

pattern discovery from user transactions, have been studied as an underlying mechanism for personalization and recommender systems [16–18]. Such techniques generally provide both a computational advantage, as well as better recommendation effectiveness, than traditional CF-based techniques, particularly in the context of click-stream data. For a recent survey of personalization based on Web usage mining see [21].

There has also been a growing body of work in enhancing collaborative filtering by integrating data from other sources such as content and user demographics [6, 20, 2, 15]. Content-oriented approaches, in particular, can be used to address the "new item problem" discussed above. Generally, in these approaches, keywords are extracted from the content of Web pages and are used to recommend other pages or items to a user, not only based on user ratings or visit patterns, but also (or alternatively) based on the content similarity of these pages to other pages already visited by the user. Keyword-based approaches, however, are incapable of capturing more complex properties of, or relationships among, objects at a deeper semantic level. Unstructured keyword-based representations often result in a substantial amount of noise resulting in reduced recommendation accuracy.

Recently, a new class of *item-based* CF algorithms has been proposed to deal with the scalability problems in user-based CF algorithms [23, 8]. Item-based CF algorithms avoid the bottleneck in user-user computations by first considering the relationships among items. Rather than finding user neighbors, the system tries to find k similar items that are rated (or visited) by different users in some similar way. Then, for a target item, predictions can be generated, for example, by taking a weighted average of the target user's item ratings (or weights) on these neighbor items. Thus, these algorithms alleviate the scalability problem that exists in user-based CF algorithms, because the similarity computations are performed in the smaller space of the items, and because often the item-item comparisons can be performed offline. At the same time, CF algorithms have been shown to achieve prediction accuracies that are comparable to or even better than user-based CF algorithms.

Item-based CF algorithms still suffer from the problems associated with data sparsity, and they still lack the ability to provide recommendations or predictions for new or recently added items. However, the item-based CF framework provides the necessary ingredients to seamlessly incorporate other sources of evidence about items (in addition to item ratings or weights). This flexibility comes from the fact that the computation of item similarities is independent of the methods used for generating predictions or recommendations, thus multiple knowledge sources, including structured semantic information about items, can be used for performing the similarity computations.

In this paper, we introduce an approach for semantically enhanced collaborative filtering in which structured semantic knowledge about items, extracted automatically from the Web based on domain-specific reference ontologies, is used in conjunction with user-item ratings (or weights) to create a combined similarity measure for item comparisons. In contrast to previous approaches to

hybrid content-collaborative systems that enhance *user based* CF [2, 15], we integrate semantic knowledge into the *item-based* CF framework. The integration of semantic similarities for items with rating (or usage-based) similarities provides two primary advantages. First, the semantic attributes for items provide additional clues about the underlying reasons for which a user may or may not be interested in particular items (something that is hidden behind the rating values in the usual context). This, in turn, allows the system to make inferences based on this additional source of knowledge, resulting in improved recommendation accuracy and coverage. Secondly, in cases where little or no rating (or usage) information is available (such as in the case of newly added items, or in very sparse data sets), the system can still use the semantic similarities to provide reasonable recommendations for users. These claims are verified by our experimental results, on two different data sets.

The rest of this paper is organized as follows. In Section 2, we provide the necessary background information on the item-based collaborative filtering framework. In Section 3, we discuss our semantically enhanced approach. In this section we first discuss the problem of ontology-based extraction of class instances in a particular domain and the structured representation of the extracted semantic attributes for items. We then present our approach for combining semantic and rating (or usage) similarity of items to generate predictions. In Section 4, we discuss the characteristics of our experimental data sets and present our experimental evaluation of the proposed approach. Finally, we conclude with a summary of our findings and some directions for future work.

2 Background on Item-Based Collaborative Filtering

In a collaborative filtering (CF) scenario, generally we start with a list of m users $U = \{u_1, u_2, \ldots, u_m\}$, a list of n items $I = \{i_1, i_2, \ldots, i_n\}$, and a mapping between user-item pairs and a set of weights. The latter mapping can be represented as a $m \times n$ matrix M. In the traditional CF domain the matrix M usually represents user ratings of items, thus the entry $M_{r,j}$ represents a user u_r's rating on item i_j. In this case, the users' judgments or preferences are explicitly given by matrix M. Collaborative filtering can also be used in the context of Web usage data. In that case, the set U may represent user sessions, some of which may belong to the same user who has visited the site multiple times. For usage data, generally, the entry $M_{r,j}$ represents an implicit weight associated with an item (e.g., page or product) i_j in a user session u_r. This weight may be binary (representing the existence or non-existence of the item in the user session), or it may be based on the amount of time spent on the particular item during the session.

For a given active user (also called the *target user*) u_a, the task of a CF system is to (1) predict $M_{a,t}$ for a given *target item* i_t which has not already been visited or rated by u_a; or (2) recommend a set of items that may be interesting to user u_a.

In user-based CF algorithms, first a set of k nearest neighbors of the target user are computed. This is performed by computing correlations or similarities between user records (rows of the matrix M) and the target user. Then, different methods can be used to combine the neighbors' item ratings (or weights) to produce a prediction value for the target user on unrated (or unvisited) items. As noted in the introduction, a major problem with this approach is the lack of scalability: the complexity of the system increases linearly as a function of the number of users which, in large-scale e-commerce sites, could reach tens of millions.

In contrast, *item-based* CF algorithms attempt to find k similar items that are co-rated (or visited) by different users similarly. This amounts to performing similarity computations among the columns of matrix M. Thus, item-based CF algorithms avoid the bottleneck in user-user computations by first considering the relationships among items. For a target item, predictions can be generated by taking a weighted average of the target user's item ratings (or weights) on these neighbor items.

2.1 Finding Similar Items (Item neighbors)

The first step in computing the similarity of two items i_p and i_q (column vectors in the data matrix M) is to identify all the users who have rated (or visited) both items. Many measures can be used to compute the similarity between items. The most common approach, when dealing with Web usage data, is to use the standard cosine similarity between two vectors:

$$sim(i_p, i_q) = \frac{\sum_{k=1}^{m} M_{k,p} \times M_{k,q}}{\sqrt{\sum_{k=1}^{m} (M_{k,p})^2 \times \sum_{k=1}^{m} (M_{k,q})^2}}$$

where $M_{k,p}$ represents the weight associated with item i_p in the session (or user) vector k.

For ratings data, however, variances in user ratings styles must be taken into account. For example, in a movie rating scenario, with a rating scale between 1 and 5, some users may give a rating of 5 to many movies they consider to be "good"; while other more "strict" raters may only give a rating of 5 to those movies they consider "perfect". To offset the difference in rating scales, the data can be normalized to focus on rating variances (deviations from the mean ratings) on co-rated items. For our purposes, when dealing with ratings data, we adapt the *Adjusted Cosine Similarity* measure introduced by Sarwar et al. [23]:

$$sim(i_p, i_q) = \frac{\sum_{k=1}^{m} (M_{k,p} - \overline{M_k}) \times (M_{k,q} - \overline{M_k})}{\sqrt{\sum_{k=1}^{m} (M_{k,p} - \overline{M_k})^2 \times \sum_{k=1}^{m} (M_{k,q} - \overline{M_k})^2}}$$

where $M_{k,p}$ represents the rating of user k on item i_p, and $\overline{M_k}$ is the average rating value of user k on all items.

2.2 Computing Predictions

After computing the similarity between items, we select a set of k most similar items to the target item and generate a predicted value for the target item. We use a *weighted sum* as follows.

$$M_{a,t} = \frac{\sum_{j=1}^{k} (M_{a,j} \times sim(i_j, i_t))}{\sum_{j=1}^{k} sim(i_j, i_t)}$$

Here, $M_{a,t}$ denotes the prediction value of target user u_a on target item i_t. Only the k most similar items (k nearest neighbors of item i_t) are used to generate the prediction.

Despite their effectiveness, item-based CF algorithms still suffer from the problems associated with data sparsity, and they still lack the ability to provide recommendations or predictions for new or recently added items. To deal with these problems, we introduce an approach for semantically enhanced collaborative filtering in which structured semantic knowledge about items, extracted automatically from the Web, is used in conjunction with user-item ratings (or weights) to create a combined similarity measure for item comparisons. This approach is discussed in the next section.

3 Using Semantic Knowledge to Enhance Collaborative Filtering

In this section, we first discuss the issue of extracting structured semantic attributes from the Web to populate instances of domain-specific ontology classes corresponding to items. We then present our approach to integrate the extracted semantic knowledge into the item-based collaborative filtering framework.

3.1 Extracting Domain Semantics from the Web

In order to obtain semantic information about items used in the collaborative filtering process, we must extract domain-level structured objects as semantic entities contained within Web pages on one or more Web sites. This task involves the automatic extraction and classification of objects of different types into classes based on an underlying reference domain ontology.

An ontology provides a set of well-founded constructs that define significant concepts and their semantic relationships. An example of an ontology is a relational schema for a database involving multiple tables and foreign keys semantically connecting these relations. Such constructs can be leveraged to build

meaningful higher level knowledge in a particular domain. Domain ontologies for a Web site usually include concepts, subsumption relations between concepts (concept hierarchies), and other relations among concepts that exist in the domain represented by the Web site. In this paper, we do not directly deal with the problems of automatic ontology acquisition and learning. Rather, we assume the existence of a pre-defined reference ontology for a specific domain based on which the semantic attributes of items can be extracted. Our goal is to use this semantic knowledge about items together with item ratings (or weights in the context of Web usage data) to create a combined similarity measure for item-based collaborative filtering.

The problem of extracting instances of the ontology classes from Web pages is an interesting problem in its own right and has been studied extensively. This process can be viewed as the classification of objects embedded in one or more Web pages into classes specified as part of a reference ontology. For example, in [10] a text classifier is learned for each "semantic feature" based on a small manually labeled data set. First Web pages are extracted from different Web sites that belong to a similar domain, and then the semantic features are manually labeled. This small labeled data set is fed into a learning algorithm as training data to learn the mappings between Web objects and concept labels. Craven et al. [7] adopt a combined approach of statistical text classification and first-order text classification in recognizing concept instances. In that study, the learning process is based on both page content and linkage information. The problems and issues related to using ontologies in the context of Web mining has been discussed in [3].

In our approach, we have used domain-specific wrapper agents that use text mining and heuristic rules to extract class and attribute instances from Web sites based on a pre-specified reference ontology. At the present time, we do not use a general ontology representation language, such as DAML+OIL [12]. Rather, we represent the ontology classes as part of the schema for a relational database. Our simple representation scheme does not take into account complex relationships among classes (such as inheritance), but is adequate for specifying the attributes associated with classes (relations). Our wrapper agents use the relational schema for classes and simple heuristics based on textual cues to extract attribute values and populate instances of these classes (tuples). In the future, we intend to extend our work by incorporating more general ontology languages that can capture (and allow reasoning with) a richer set of structural relationships among classes and objects. The implementation details of the wrapper agents is beyond the scope of the present work and will be discussed elsewhere.

As an example, let us consider a movie Web site such as the Internet Movie Database (www.imdb.com). This Web site includes a collection of pages containing information about movies, actors, directors, etc. A collection of pages describing a specific movie might include attribute information such as the movie title, genre, actors, director, etc. These represent the attributes associated with a class that represents movies in our reference ontology. A domain ontology for this site may contain the classes **Movie**, **Actor** and **Director** along with their

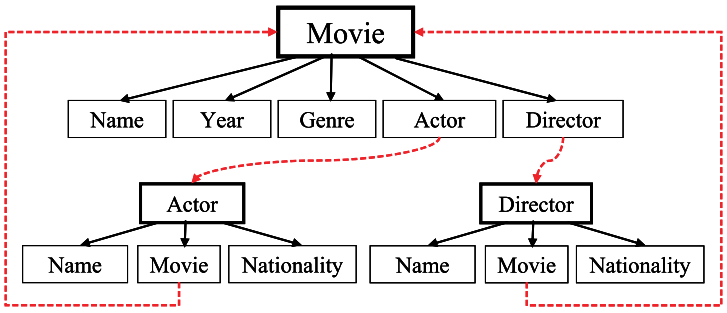

Fig. 1. Portion of the ontological representation for a movie Web site

attributes. In our representation, some of the attributes represent properties of a given class and others represent reference slots corresponding to other classes. For instance, the "Actor" attribute of the **Movie** class represents a reference to the class (relation) **Actor** and, in the relational representation, is specified as a foreign key in the **Movie** relation. Figure 1 depicts the class **Movie** and its attributes. An actor or director's attribute information may include name, filmography (a set of movies), gender, nationality, etc. The dotted arrows in attributes such as "Actor" and "Director" indicated that they represent references to other classes in the ontology. The collection of Web pages in the site represent a group of embedded objects that are the instances of these classes.

In order to facilitate the computation of item similarities, generally, the extracted class instances will need to be converted into a vector representation. In our case, the values of semantic attributes associated with class instances are collected into a relational table whose rows represent the n items, and whose columns correspond to each of the extracted attributes. Additional preprocessing tasks, such as normalization and discretization (for continuous attributes), can be performed on the data in order to provide a uniform representation. This process generally results in the addition of attributes, for example, representing different intervals in a continuous range, or representing each unique discrete value for categorical attributes in the original data. The final result is a $n \times d$ matrix S, where d is the total number of unique semantic attributes. We call this matrix the *semantic attribute matrix*.

3.2 Integrating Semantic Similarity with Collaborative Filtering

As noted earlier, the item-based CF framework provides a computational advantage over user-based approaches, since item similarities can be computed offline, prior to the online task of generating recommendations. But, this framework also provides another important advantage. Since the computation of item similarities is independent of the methods used for generating predictions or recommendations, other sources of evidence about items (in addition to item ratings or weights) can be used for performing the similarity computations.

The integration of semantic similarities for items with rating (or usage-based) similarities provides two primary advantages. First, the semantic attributes for items provide additional clues about the underlying reasons for which a user may or may not be interested in particular items (something that is hidden behind the rating values in the usual context). This, in turn, allows the system to make inferences based on this additional source of knowledge, possibly improving the accuracy of recommendations. Secondly, in cases where little or no rating (or usage) information is available (such as in the case of newly added items, or in very sparse data sets), the system can still use the semantic similarities to provide reasonable recommendations for users.

In the following we describe our approach for integrating semantic similarities into the standard item-based collaborative filtering framework. Our approach involves first performing latent semantic analysis on the semantic attribute matrix obtained using the process described in Section 3.1. This is necessary in order to reduce noise and to collapse highly correlated attributes. We then compute item similarities, both based on the reduced semantic attribute matrix, as well as based on the user-item ratings (or usage) matrix. Finally, we use a combined similarity measure, as a linear combination of the two similarities to perform item-based collaborative filtering.

Using Latent Semantic Analysis on Semantic Attributes. Latent Semantic Indexing (LSI) [4] is a dimensionality reduction technique which is widely used in information retrieval (IR). Many IR applications have shown that performing latent semantic analysis, including in document indexing, can improve the accuracy of information retrieval. Given a term-document frequency matrix, LSI is used to decompose it into two matrices of reduced dimensions and a diagonal matrix of singular values. Each dimension in the reduced space is a latent variable (or factor) representing groups of highly correlated index terms. Reducing the dimensionality of the original matrix reduces the amount of noise in the data as well as its sparsity, thereby, improving retrieval based on the computation of similarities between the indexed documents and user queries. Here we apply this idea to create a reduced dimension space for the semantic attributes associated with items.

Singular Value Decomposition (SVD) is a well known technique used in LSI to perform matrix decomposition. In our case, we perform SVD on the semantic attribute matrix $S_{n\times d}$ by decomposing it into three matrices:

$$S_{n\times d} = U_{n\times r} \bullet \Sigma_{r\times r} \bullet V_{r\times d}$$

where U and V are two orthogonal matrices; r is the rank of matrix S, and Σ is a diagonal matrix of size $r \times r$, where its diagonal entries contain all singular values of matrix S and are stored in decreasing order. One advantage of SVD is that it provides the best lower rank approximation of the original matrix S [4]. We can reduce the diagonal matrix Σ into a lower-rank diagonal matrix $\Sigma_{k\times k}$ by only keeping k ($k < r$) largest values. Accordingly, we reduce U to U' and

V to V'. Then the matrix $S' = U' \bullet \Sigma' \bullet V'$ is the rank-k approximation of the original matrix S.

In the above process, U' consists of the first k columns of the matrix U corresponding to the k highest order singular values. In the resulting semantic attribute matrix, S', each item is, thus, represented by a set of k latent variables, instead of the original d attributes. This results in a much less sparse matrix, improving the results of similarity computations, as well as the computational cost associated with the process. Furthermore, the generated latent variables represent groups of highly correlated attributes in the original data, thus potentially reducing the amount of noise associated with the semantic information. As we will illustrate in the next section, performing latent semantic analysis on the semantic space, generally leads to substantial gains in prediction accuracy based on the semantic attributes.

Predictions Based on a Combined Similarity Measure. The semantic similarity measure $SemSim(i_p, i_q)$, for a pair of items i_p and i_q, is computed using the standard vector-based cosine similarity on the reduced semantic space. This process can be viewed as multiplying the matrix S' by its transpose and normalizing each corresponding row and column vector by its norm. This results in a $n \times n$ square matrix in which an entry i, j corresponds to the semantic similarity of items i and j.

Similarly, we compute item similarities based on the user-item matrix M. As noted in Section 2, in the case of usage data, we use the cosine similarity measure. In the case of ratings data (such as movie ratings) we employ the adjusted cosine similarity in order to take into account the variances in user ratings. We denote the rating (or usage) similarity between two items i_p and i_q as $RateSim(i_p, i_q)$.

Finally, for each pair of items i_p and i_q, we combine these two similarity measures to get $CombinedSim$ as their linear combination:

$$CombinedSim(i_p, i_q) = \alpha \cdot SemSim(i_p, i_q) + (1 - \alpha) \cdot RateSim(i_p, i_q)$$

where α is a *semantic combination parameter* specifying the weight of semantic similarity in the combined measure. If $\alpha = 0$, then $CombinedSim(i_p, i_q) = RateSim(i_p, i_q)$, in other words we have the standard item-based filtering. On the other hand, if $\alpha = 1$, then only the semantic similarity is used which, essentially, results in a form of content-based filtering. Finding the appropriate value for α is not a trivial task, and is usually highly dependent on the characteristics of the data. We choose the proper value by performing sensitivity analysis for particular data sets in our experimental section below.

In order to compute predicted ratings or recommendations, we use the weighted sum approach discussed in Section 2. Specifically,

$$M_{a,t} = \frac{\sum_{j=1}^{k} (M_{a,j} \times CombinedSim(i_j, i_t))}{\sum_{j=1}^{k} sim(i_j, i_t)},$$

where, $M_{a,t}$ denotes the prediction value of target user u_a on target item i_t.

4 Experimental Evaluation

In this section we compare the semantically enhanced and standard item-based collaborative filtering in the context of two different data sets. In the first case, we focus our attention to the traditional context in which collaborative filtering is used, namely that of item ratings. For this purpose we choose the domain of movies and user's ratings of these movies. Secondly, we apply our approach to Web usage data. Specifically, we have chosen a real estate Web site containing information about various residential properties. While these data sets are quite different, the experimental results in this section demonstrate that the integrated approach yields advantages both in terms of improving accuracy, as well as in resolving some of the shortcomings associated with traditional approaches.

In each case, the data set was divided into random training and test sets. The training sets were used to build the models while the test sets were used to generate and evaluate recommendations. To assure statistical accuracy, this process was repeated five times for different random partitionings of the data. Unless otherwise specified, all of the results reported in this section represent averages over the five folds.

4.1 Data Sets and Evaluation Metrics

For the movie data set we used the ratings data from the MovieLens recommendation system (www.movielens.org). This data set contains 100,000 ratings on 1682 movies from 943 users. Each user has rated 20 or more movies with a rating scale of 1 to 5. We used our own wrapper agent to extract movie instances from the Internet Movie Database (www.imdb.com) based on the movie ontology depicted in Figure 1. Specifically, each instance was populated with semantic attributes, including movie title, release year, director(s), cast, genre, and plot.

The extracted instances were then converted into a binary table in standard spreadsheet format, where each row represents a movie, and each column represents a unique attribute value. For attributes involving continuous data types (such as "price" and "year") we performed discretization to generate a set of intervals as attributes. Similarly, for attributes involving a concept hierarchy, each concept node was represented as a unique attribute. This process resulted in a table representing each movie as an attribute vector with 2762 dimensions. Prior to computing the semantic similarity among movies, singular value decomposition was performed on the data, using different SVD dimensions, resulting in the corresponding semantic similarity matrices. The generated similarity matrices where then used in our experiments along with the rating similarities among movies, computed from the original ratings data.

To measure the accuracy of the recommendations we computed the standard *Mean Absolute Error* (MAE) between ratings and predictions in the test data

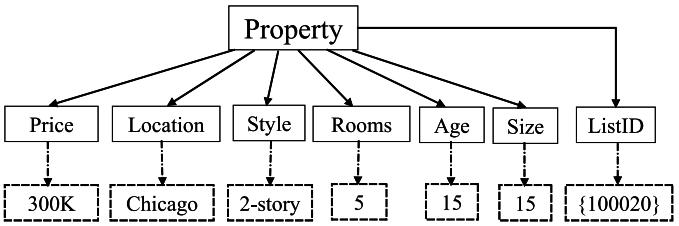

Fig. 2. Portion of the ontology for the class "Property" in the real estate Web site

sets. Specifically, given the set of actual/predicted rating pairs $\langle a_i, p_i \rangle$ for all the n movies in the test set, the MAE is computed as:

$$MAE = \frac{\sum_{i=1}^{n} |a_i - p_i|}{n}.$$

Note that lower MAE values represent higher recommendation accuracy. In this case, the ratings are based on a discrete scale of 1 (lowest) to 5 (highest). Thus, the maximum possible value for MAE is 4 (indicating a maximum possible error on all predictions).

In the case of the real estate data, we started with the raw Web usage data from the server logs of a local affiliate of a national real estate company. The primary function of the Web site is to allow prospective buyers visit various Web pages containing information related to some 300 residential properties. The portion of the Web usage data during the period of analysis contained approximately 24,000 user sessions from 3800 unique users. The preprocessing phase for this data was focused on extracting a full record for each user of properties she visited. This required performing the necessary aggregation operations pageviews in order to treat a property as the atomic unit of analysis. In addition, the visit frequency for each user-property pair was recorded, since the number of times a user comes back to a property listing is a good measure of that user's interest in the property. Finally, the data was filtered to limit the final data set to those users that had visited at least three properties. In our final data matrix, each row represented a user vector with properties as dimensions and visit frequencies as the corresponding dimension values.

To automatically extract semantic information about the properties, we used a reference ontology for the domain depicted in Figure 2. In this case, our ontology only contained a single class called "property." The figure only shows a subset of the attributes associated with "property" that were used for computing semantic similarities. An example of an instance of this class is also depicted in Figure 2 (dotted arrows show the mapping between each attribute and the corresponding attribute value in the extracted instance). Using a wrapper agent based on this reference ontology, the attribute values for each property instance were extracted directly from pages related to that property on the Web site. The discretization and normalization process described above for the movie data was also applied in this case resulting in final set of 120 unique attribute dimensions

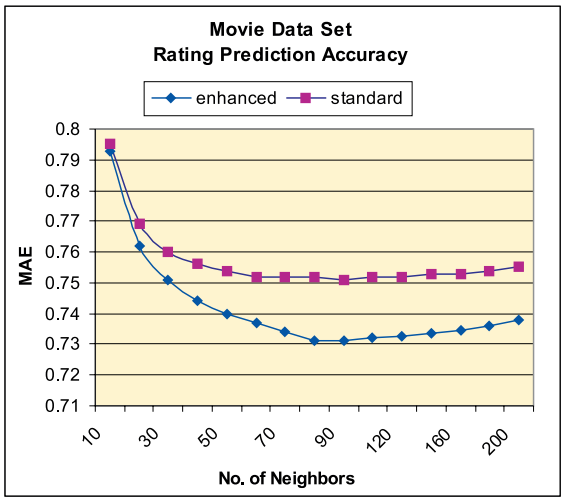

Fig. 3. Prediction accuracy for semantically enhanced recommendations v. standard item-based collaborative recommendations

for each property vector. We then applied singular value decomposition to generate different semantic similarity matrices that were used in our experiments.

In contrast to the movie data set, this usage data does not involve item ratings. Thus, the standard MAE measure is not the appropriate approach for determining the accuracy of predictions. Instead we use the notion of *hit ratio* in the context of top-N recommendations. For each user, we randomly held one visited property as test data and used the rest as training data. The recommendation algorithm generates the top N recommended properties in the test set. If the previously held property appears in the recommendation set, this is considered a *hit*. We defined the Hit Ratio as the total number of hits divided by the total number of users in the test set.

It should be noted that the hit ratio increases as the value of N (number of recommendations) increases. Thus, in our experiments, we pay especial attention to a smaller number of recommendations (between 1 and 10) that result in good hit ratios.

4.2 Experiments with Movie Ratings Data

Figure 3 depicts the prediction accuracy of our semantically enhanced recommendations in contrast to those produced by standard item-based collaborative filtering. Here the MAE has been plotted with respect to the number of neighbors (similar items) in the k-nearest-neighbor algorithm. In both cases, the MAE converges between 80 and 100 neighbors, however, the semantically enhanced approach results in an overall improvement in accuracy.

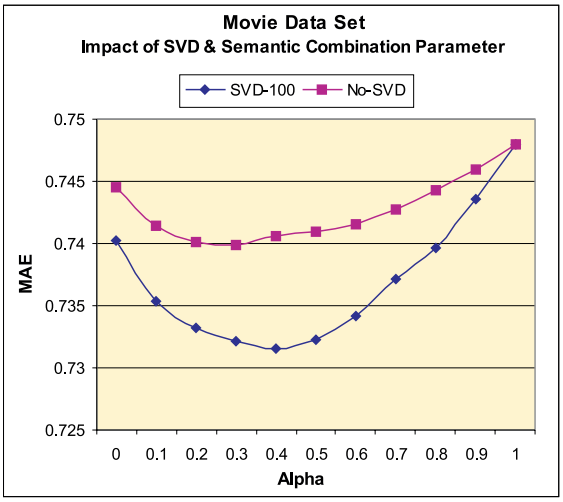

Fig. 4. Impact of the semantic combination parameter and SVD-based dimensionality reduction on recommendation accuracy

A more telling picture emerges when we compare the range of values for the semantic combination parameter α. Recall that α is the parameter determining the degrees to which the semantic and rating similarities are used in the generation of neighbors. When $\alpha = 0$, then only semantic similarity among items is used, while $\alpha = 1$ represents the other side of the spectrum where only rating similarity is used (i.e., standard item-based recommendations). Figure 4 serves two purposes. First, it shows the impact of α on MAE, and secondly, it shows the impact of performing singular value decomposition (in this case, 100 dimensions) on the semantic data prior to computing similarities.

Applying SVD provides a two-fold advantage. On the one hand, SVD generally results in much better computational performance during tasks such as similarity computations or clustering. On the other hand, as clearly indicated by these results, it results in a general improvement in recommendation accuracy (most likely due to a reduction in noise). In the SVD case, the optimum value of α is around 0.40 which is also the point at which performing SVD has the largest impact. Note that at $\alpha = 1$, results for SVD-100 and no SVD are the same, since in that case the semantic similarity matrix is not taken into account. Interestingly, the results also show that in this data set using only semantic attributes ($\alpha = 0$) results in recommendations whose quality are in par with (or better) than recommendations based on rating similarities. However, it is clear that the combination of semantic and rating similarities provides an advantage over both of these boundary conditions.

As noted earlier, one of the problems associated with traditional collaborative filtering algorithms emanate from the sparsity of data sets to which they are applied. This sparsity has a negative impact on the accuracy and predictability

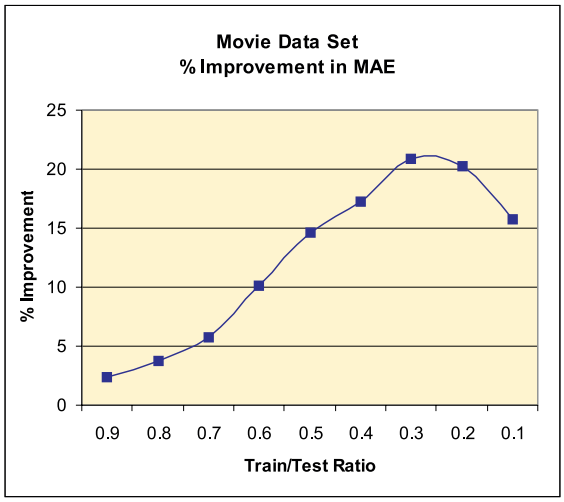

Fig. 5. Improvement in MAE for different test/train ratios (using 100 SVD dimensions and $\alpha = 0.4$)

of recommendations. This is one area in which, we believe, the integration of semantic knowledge with ratings data can provide significant advantage. To test this hypothesis, we created multiple training/test data sets in which the proportion of the training data to the complete ratings data set was changed from 90% to 10%. These proportions have a direct correspondence with the level of sparsity in the ratings data. In the case of each of the combination parameter values, we created five random training and test data sets and computed average MAE's over the five folds. We then computed the average improvement in MAE achieved by the semantically enhanced method over the standard item-based CF approach.

Figure 5 depicts these results for the SVD-100 data using a combination parameter $\alpha = 0.4$. While the overall recommendation accuracy drops as the proportion of training data is reduced (not shown), the results indicate that, generally, for sparser data sets, the semantic approach achieves larger improvements. As might be expected, this improvement starts to converge to 0 for very sparse data sets. This is because for very small training sets, neither approach can generate a reasonable number of recommendations. However, for up to a training ratio of 30%, the semantic approach provides improvements of up to 20% in MAE scores.

As a final experiment with the movie data set, we focused our attention on another common problem with CF-based approaches, namely, the "new item" ("cold start") problem: since there are no ratings for new items, standard item-based algorithm cannot find item neighbors using rating similarity and fail to give predictions. Our goal here was to determine the degree to which semantic

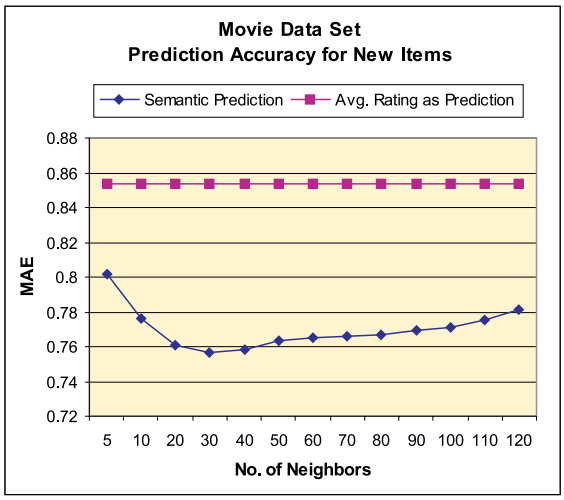

Fig. 6. Using semantically enhanced predictions for new (previously unrated) items

information from the domain can help produce accurate recommendations in the absence of any available ratings data for new items.

To achieve this goal we chose all movies which only received one rating and held these ratings as the test data. The actual movies in the selected data set were predominantly those that were very recently released (relative to the last date captured by the data). Thus, the sample closely modeled the conditions under which newly added items are considered for recommendation. In the training data, these movies received no ratings at all, and thus they were considered to be "new items". We compared our algorithm to a baseline algorithm, in which each user's average rating from training data was used as the prediction for "new items" in test data. These results are depicted in Figure 6. They show that, at all neighbor size levels, our algorithm can provide more accurate predictions than the baseline case.

4.3 Experiments with Real Estate Usage Data

As our first experiment with the Reality data, we compared the hit ratio of the semantically enhanced approach at different combination parameters for both the complete data set and the 40-dimension SVD data set. These results are depicted in Figure 7. In this case we only focused on the Top 10 recommendations generated by the algorithm.

The results suggest similar conclusions to those observed in the movie data set. First, in general, singular value decomposition has an even more dramatic impact in this case; more so when the focus is shifted to the semantic information (α close to 0) as opposed to usage data. In fact, we see that in this data set, without performing SVD on the semantic attribute matrix, the combined

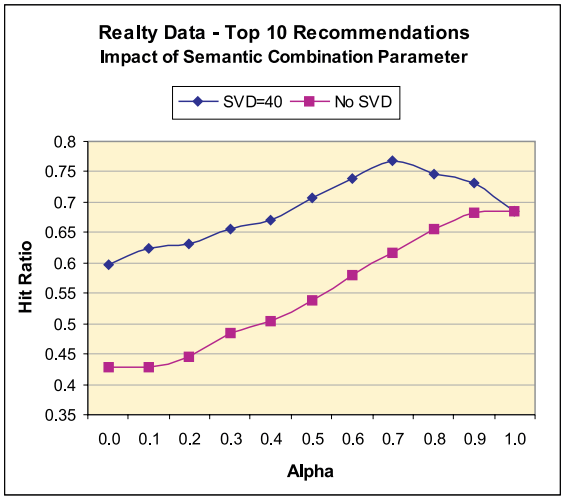

Fig. 7. Impact of semantic combination parameter for the top 10 recommendations in the real estate usage data

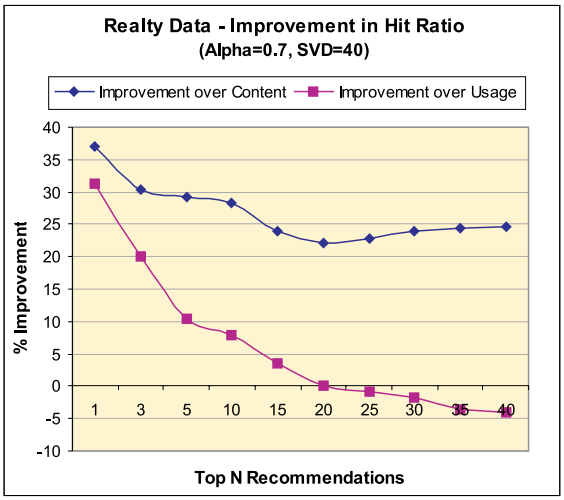

Fig. 8. Improvement of the semantically enhanced recommendations over content-only and usage-only recommendations

approach does not improve accuracy when compared to pure usage-based recommendations. This may be an indication that many different attributes contribute to the type of property in which visitors show interest. Applying SVD results in a smaller number of latent factors by combining multiple attributes. These factors, individually, may be more predictive in determining user interests than the more fine grained attributes. As can be seen, with SVD the semantic approach results in significant improvements over both usage-only and content-only recommendations, particularly at a combination parameter $\alpha = 0.7$.

Next, we measured the hit ratio improvement achieved by our algorithm (with semantic combination parameter $\alpha = 0.7$), over the two boundary cases when only usage-based similarity ($\alpha = 1$) or only semantic similarity ($\alpha = 0$) are used to generate recommendations. Figure 8 depicts these results. The combined similarity measure achieved between 20% to 37% improvement over the semantic-only recommendations (i.e., over pure content-based filtering). In the case of usage-based recommendations, we observe that with recommendation sets of size less than 20, the combined approach always achieved better Hit Ratio. The improvement is particularly significant for small values of N. Indeed, in real situations, we are interested in few, but accurate recommendations, and this is precisely where the semantically enhanced approach seems to provide the most advantage.

5 Conclusions and Future Work

In this paper we have extended the item-based collaborative filtering framework by integrating structured semantic information about items for similarity computations. We have used domain-specific reference ontologies to automatically extract such features from the Web and populate class instances. Our enhanced similarity measure combines domain-based semantic item similarities with item similarities based on the user-item mappings. Our experimental results show that the semantically enhanced approach improves the prediction accuracies, while maintaining the computational advantages of item-based CF. In the context of Web usage and e-commerce data, the improvements are even more significant, particularly when focusing on a small number of recommendations.

The application of latent semantic analysis to the extracted semantic features, which reduces noise in the data, further improves the results when the hybrid approach is compared to usage-only or semantic-only recommendations. Furthermore, we have experimentally shown that, for new, unrated items, our approach can produce reasonably accurate recommendations, thus alleviating the "new item problem" associated with standard collaborative filtering. Our experiments also suggest that the integrated approach provides better quality predictions in the face of very sparse ratings or usage data.

An interesting area of current and future work is to use the characteristics of the domain together with machine learning techniques to automatically determine the semantic combination parameter (i.e., the degree to which the semantic similarity is combined with the item similarities based on ratings or usage).

We will also further study the impact of using other approaches for measuring semantic similarities which take into account the structure of the underlying domain ontologies. Of particular relevance in this context is the work of Ganesan et al. [9] on using hierarchical structures in computing similarities, and that of Hotho et al. [13] on ontology-based text clustering.

References

1. C. C. Aggarwal, J. L. Wolf, and P. S. Yu. A new method for similarity indexing for market data. In *Proceedings of the 1999 ACM SIGMOD Conference*, Philadelphia, PA, June 1999.
2. C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the the 15th National Conference on Artificial Intelligence (AAAI 98)*, Madison, WI, July 1998.
3. B. Berendt, A. Hotho, and G. Stumme. Towards semantic web mining. In *Proceedings of the First International Semantic Web Conference (ISWC02)*, Sardinia, Italy, June 2002.
4. M.W. Berry, S.T. Dumais, and G.W. O Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.
5. D. Billsus and M.J. Pazzani. Learning collaborative information filters. In *Proceedings of the International Conference on Machine Learning*, Madison, WI, 1998.
6. M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, California, August 1999.
7. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1-2):69–113, 2000.
8. M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):1–34, 2004.
9. P. Ganesan, H. Garcia-Molina, and J. Widom. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems*, 21(1):63–94, 2003.
10. R. Ghani and A. Fano. Building recommender systems using a knowledge base of product semantics. In *Proceedings of the Workshop on Recommendation and Personalization in E-Commerce, at the 2nd Int'l Conf. on Adaptive Hypermedia and Adaptive Web Based Systems*, Malaga, Spain, May 2002.
11. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, Berkeley, CA, August 1999.
12. I. Horrocks. Daml+oil: A reasonable web ontology language. In *Proceedings of the 8th International Conference on Extending Database Technology*, pages 2–13, Prague, Czech Republic, March 2002. Springer-Verlag.
13. A. Hotho, A. Maedche, and S. Staab. Ontology-based text clustering. In *Proceedings of the IJCAI-2001 Workshop Text Learning: Beyond Supervision*, Seattle, WA, August 2001.
14. J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3), 1997.

15. P. Melville, R.J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering. In *Proceedings of the SIGIR2001 Workshop on Recommender Systems*, New Orleans, LA, September 2001.
16. B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.
17. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd ACM Workshop on Web Information and Data Management (WIDM01)*, Atlanta, Georgia, November 2001.
18. B. Mobasher, H. Dai, and M. Nakagawa T. Luo. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6:61–82, 2002.
19. M. O'Conner and J. Herlocker. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR Workshop on Recommender Systems*, Berkeley, CA, August 1999.
20. M. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.
21. D. Pierrakos, G. Paliouras, C. Papatheodorou, and C.D. Spyropoulos. Web usage mining as a tool for personalization: A survey. *User Modeling and User-Adapted Interaction*, 13:311–372, 2003.
22. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems–a case study. In *Proceedings of the WebKDD 2000 Workshop at the ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD'00)*, August 2000.
23. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International WWW Conference*, Hong Kong, May 2001.
24. B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommender algorithms for e-commerce. In *Proceedings of the 2nd ACM E-Commerce Conference (EC'00)*, Minneapolis, MN, October 2000.
25. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating 'word of mouth'. In *Proceedings of the Computer-Human Interaction Conference (CHI95)*, Denver, CO, May 1995.
26. J. Srivastava, R. Cooley, M. Deshpande, and P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.