Ontology-based Top-N Recommendations on New Items with Matrix Factorization

Haomin Cui[a], Ming Zhu[a], Shijia Yao[a]
[a] Department of Automation, University of Science and Technology of China, Hefei 230027, China
Email: hmincui@gmail.com
Email: mzhu@ustc.edu.cn
Email: yaosj@mail.ustc.edu.cn

***Abstract*— Collaborative Filter is proved to be effective in recommendations and widely used in the recommender system for online stores. The mechanism of this method is to find similarities among users in rating score. The item can be recommended based on the similar user's choice. The calculation of user similarities is based on distance metrics and vector similarity measures. However, the effect of CF methods is limited by several problems, such as the new item problem and how to recommend the items in the long-tail. The data sparsity, which means fewer scores in user rating matrix, can lead to difficulties in finding a relationship among users for recommendations. It is particularly important to design new similarity metrics which is based on the inherent relationship between items rather than rating score by users. In this paper, we introduce an approach using ontology-based similarity to estimate missing values in the user rating matrix. To accommodate different features of items, we investigate several kinds of metrics to estimate the similarity of item ontology, such as Tversky's similarity, Spearman's rank correlation coefficient, and Latent Dirichlet Allocation. The missing rating score was filled by the mechanism based on the similarity of the item ontology. With the new rating matrix, the original CF method could get better performance in recall. Experiments using Hetrec'11 dataset were carried out to evaluate the proposed methods using Top-N recall metrics. The results show the effect of the proposed method compared with state-of-the-art approaches when applied to new item cold start and long-tail situations.**

***Index Terms*— Ontology Similarity, Recommender System, Matrix Factorization, Data Sparsity**

I. Introduction

RECOMMENDER/ systems (RS) always perform as the medium between users and providers in online stores. As the development of online application, large amount of data including new movies, music, books and other media contents need to be discovered. From the user's perspective, various tastes develop a needing for filtering useful items suitable for user's own interest.

Recommender system mainly contains two categories [1]: Content-based Filtering and Collaborative Filtering. Content-based systems make a recommendation based on historical choice of users to items, while the collaborative filtering systems find neighbor users and make recommendation based on their selection of items to target users. Many successful recommender system is used Collaborative Filtering (CF) method, which is based-on user-by-item matrix to predict user's choice. Neighborhood methods [2] and Latent factor models [3] are two basic disciplines of CF.

However, as previous works [4]–[6] have revealed that CF suffers from matrix sparsity and cold-start problem, which strongly affect the performance in recommendation. Content-based filtering only use attributions of items and have natural advantages for sparsity and cold-start problems. Hybrid models have proposed in the past several years.

In this paper, we propose ontology-based semantic similarity to estimate the missing value in user-by-item matrix, in order to address the data sparsity problem. Ontological features indicate inherent relations among items. The similarity between items is dependent on their properties, that is, common features tend to increase similarity and non-common ones tend to diminish it [7].

Experiments using Hetrec'11 dataset [8] is revealed the effective of proposed method compared with state-of-art methods.

In summary, the contributions of this paper are as follows:

1) We shows that filling missing value with non-zeros in user-item matrix is significant for Top-N recommender system.
2) We proposed a novel ontology-based similarity as for missing rating value estimate.
3) We use serval experiments to show the effective and efficient for proposed method.

The rest of the paper is organized as follows. In Section II, we describe the related works on collaborative filtering recommend system. Section III reviews the preliminaries that are referred to in this paper. In Section IV, we propose an ontology-based similarity. Performance evaluation and conclusions are described in Section V and Section VI, respectively.

Manuscript received September 10, 2013; revised January 2, 2013; accepted April 16, 2013.

This work was supported in part by National Key Technologies R&D Program of China (GrantNo.2011AA01A107) and the "Strategic Priority Research Program" of the Chinese Academy of Sciences(Grant No. XDA06030900).

doi:10.4304/jsw.9.8.2026-2032

II. Related Works

A. Top-N Recommender

The Top-N recommender system has been discussed widely in recent years. The goal of this kind system is to find niche products which stay hidden in long tail of whole products. In [9], the author stated that lack of reliable information and being unreachable by any sequence of recommendations are two major problems for recommending a niche product. In practice, traditional collaborate filtering suffers from the new item problem. New items cannot be recommended as old items that have been selected or rated by many users.

In recent years, many works incorporate contextual information into recommendation process. A number of context-aware recommender systems have been proposed. In [10], contextual information is added to the standard dimensions related to users and items. In [11], the contextual values are used as virtual items together with standard ones. In [12], a novel pre-filtering technique for context-aware CF called item splitting is proposed. In this approach, the ratings of specified items are split, according to the value of an item-dependent contextual condition. Each split item generates two fictitious items that are used in the prediction algorithm instead of the original one. In [13], a post-filters recommendations on the basis of contextual information are proposed. The mined rules are utilized as to identify the most significant correlations among context and item characteristics. The predictions are filtered to provide contextualized recommendations.

B. Cold Start

In recommender system, the insufficient number of the transactions and feedback data lead to the sparsity problem, which confined the performance of the collaborative filtering [14]. To solve data sparsity problem, many recommended algorithms are used to find underlying correlation between item [15] and user clusters [16].

Hybrid algorithms have been brought to the scientific attention during the last few years. In [17], mainly classes in hybrid recommender systems are defined as follows: mixed, switched, weighted, feature-augmentation and meta-level hybrids. Weighted algorithms compute a linear combination of the ratings predicted by two (or more) recommender algorithms. This method can be used on implicit datasets among all hybrid solutions. A similar approach has been proposed in [18] by Mobasher et. al. This method linearly combines item-to-item similarity results generated by different recommender algorithms. In [19], a multi-aspect probabilistic model which is utilized to compute the probability that an item is liked or not by a user is proposed. However, this kind of method is completely different from the collaborative filtering system and not properly scale with the number of users and items [20].

C. Ontology-Based Recommender

Several research studies have focused on ontology and recommender system. In [21], an ontology containing information automatically extracted from departmental databases available on the web. The ontology is used to address the recommender systems cold-start problem. In [22], a prototype web-based CCBR system has been proposed using ontology-based semantic metadata in RDF. In [23], news contents and user preferences are described in terms of concepts appearing in a set of domain ontologies. Based on the similarities between item descriptions and user profiles, and the semantic relations between concepts, content-based and collaborative recommendation models are supported by the system. In [24], most of the ontology-based approaches are classified. A new ontology-based measure is relying on the exploitation of taxonomical features.

Different from the previous works, we proposed a three-step recommender system. We are using ontology similarity between new items and existing ones for predicting rating values to attack the sparsity of user-item rating matrix. In this way, the traditional collaborative filtering method could performance better founded on more rating values. After Collaboration filtering process, the rough recommender list will be clustered based on the ontology similarity and improve the diversity of the recommender system, as for optional step.

III. Preliminaries

In this section, we describe several state-of-the-art algorithms for top-N recommendation briefly. For more detail about these preliminaries, see [2], [25], [26].

A. Popularity-based

MovieAvg and TopPop are two mainly recommended methods based on the popularity of items. MovieAvg uses the average rating as metric for movies, and it recommends the top-N highest rating. For TopPop method, the ranking list is built on the number of ratings. Top-N items with the highest popularity will be recommended to all users. Without considering the individual interest difference, the recommended result of these methods is a constant list to every user. However, it indicates the baseline of recommendation.

B. Neighborhood Collaborative Filtering

Neighborhood Collaborative Filtering used the similarity between items or users. For user-based NNCF, the similarity between users has been measured as the cosine similarity. The user profile can be described by the rating history or demographical information. The similarity rank list is ordered by similarity score, and the top-N user's rating will be used for predicting, as (1):

$$\hat{r}_{ui} = \sum_{u' \in \hat{U}_{knn}} r_{u'i} \cdot sim_{uu'} \tag{1}$$

where $\hat{U}_{knn}$ is the top-k most similar users, and the $sim_{uu'}$ is the cosine similarity of rating vectors between u and u'. On the other hand, as users tend to rate similar items similarly, item-based NNCF uses the user's own ratings on similar items for predicting unknown rating, as equation (2) shows.

$$\hat{r}_{ui} = \sum_{j \in D^k(u,i)} r_{uj} \cdot sim_{ij} \tag{2}$$

According to [2], normalization shows a little effect on the recommended result. Non-normalized method becomes easy to use in practice application and performs well.

C. PureSVD

PureSVD is a collaborative filtering method based on matrix factorization. In [25], it shows high recall in long-tail recommendation. Built on singular value decomposition (SVD), the user-rating matrix is factorized into d-dimensional matrices as equation (3) shows.

$$R = U \cdot \mathbf{\Lambda} \cdot Q^T \tag{3}$$

The rating prediction is computed as equation (4) shows.

$$\hat{r}_{ui} = \boldsymbol{r}_u \cdot \boldsymbol{Q} \cdot \boldsymbol{q}_i^T \tag{4}$$

The dimension d is the number of singular values for $\boldsymbol{R}$, which should be decided manually. Notice that the symbol $\hat{r}_{ui}$ is not exactly a valid rating value, but an association measure between user u and item i, which is used for ranking.

D. Latent Semantic Analysis

Latent Semantic Analysis used SVD to reduce item-content matrix (ICM) dimensionality and to find the underlying relationships between items. The original ICM is approximated as $\boldsymbol{W}_l = \boldsymbol{Z} \cdot \mathbf{\Lambda} \cdot \boldsymbol{Y}^T$. Items can be described by l latent semantic features $\boldsymbol{B} = \boldsymbol{Y} \cdot \mathbf{\Lambda}$. Rating prediction is computed as $\hat{r}_{ui} = \boldsymbol{r}_u \cdot \boldsymbol{B}$. It belongs to content-based methods and has a natural advantage for new items recommendation. For more detailed information about this method, see [26].

IV. The Ontology-Based Similarity

In this section, we formally describe the ontology-based similarity to our work. Feature-based measure is one of mainly categories for ontological similarity assessment [24]. Fig.1 shows the ontology structure in the field of movie. Actor, director, genres and tag are basic features for typical movie ontology. Each feature has child features such as the rank of the actor, the weight of genres and the latent semantic relation among tags. In practice, different features may have distinct impact on similarity. We proposed several similarity measures as follows:

A. Tversky's Similarity

For two sets of features, the Tversky's model of similarity is the basic formation, which considers the similarity of two items can be computed as a function of common and different features of items [7]. For sets X and Y of

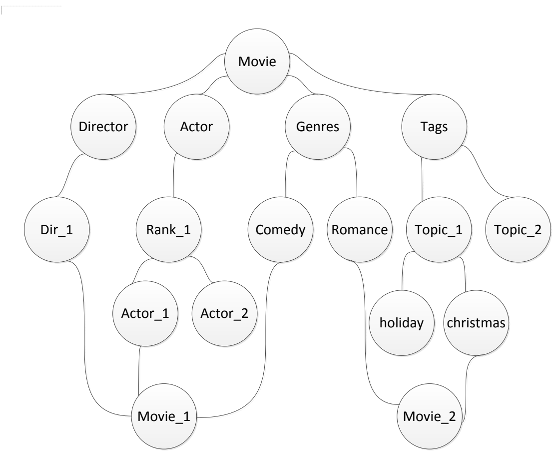


Figure 1. The basic ontology structure of Movie.

item features, the Tversky index is a function of $X \cap Y$, $X - Y$ and $Y - X$, as equation (5) shows:

$$TSim(X,Y) = \frac{|X \cap Y|}{|X \cap Y| + \alpha|X - Y| + \beta|Y - X|} \tag{5}$$

where α and β are parameters that weight the contribution of each component. We put forward this the weighted version of Tversky's similarity, as equation (6) shows:

$$TWSim(X,Y) = \frac{W_{|X \cap Y|}}{W_{|X \cap Y|} + \alpha W_{|X-Y|} + \beta W_{|Y-X|}} \tag{6}$$

The weighted coefficients can be calculated by TF-IDF technique for terms.

B. Rank Similarity

Some features have little similarity literally, such as actors in movie ontology. They may be located in a rank list ordered by importance or occurring frequency. Based on this rank list, we could estimate the possible similarity between features irrelative with the literal sense by Spearman's rank correlation coefficient. Suppose features X_i and Y_i of the movie i, j are converted to ranks x_i, y_i, the rank similarity $RankSim(i,j)$ is shown in equation (7):

$$RankSim(i,j) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \tag{7}$$

For two lists in the ascending order, the two similar rank lists will have a higher Spearman's coefficient.

C. Semantic Similarity

Web systems such as Flickr and IMDB may allow people to create and manage tags to annotate and categorize content. Like keywords of documents, tags may not only be literally common but also have semantic similarity. Different from genres, individual difference of vocabulary loads to variable tag annotations, while the latent topic behind tags may be the same. There are various kinds of methods for latent semantic analysis in information

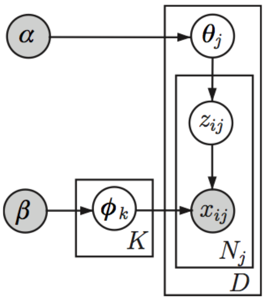


Figure 2. The graphical model for LDA.

retrieval circles. Latent Dirichlet allocation (LDA) is one of a powerful method for successfully using in tagging system [27]. The items can be represented as a set of discrete topics. The tags reflect a common vocabulary to describe the topics. The process of LDA is to find the mixture of topics for each items, as equation (8) shows

$$P(t_i|d) = \sum_{j=1}^{Z}(P(t_i|z_i = j)P(z_i = j|d)) \quad (8)$$

where $P(t_i|z_i = j)$ is the probability of t_i in topic j and the $P(t_i|d)$ is the probability of the ith tag for the document d. LDA estimates the topic-term distribution $P(t|z)$ and the document-topic distribution $P(z|d)$ with a fixed number of topics. A Markov chain Monte Carlo (MCMC) method called after Gibbs samplings is utilized to perform inference. Fig.2 shows the graphical model for LDA, where x_{ij} is the tags. α is the parameter of the Dirichlet prior on the per-document topic distributions, and β is the parameter of the Dirichlet prior on the per-topic word distribution. Both are the hyper parameters for the Dirichlet priors, serving as smoothing for the counts.

Based on this method, different tags on the same topics can be related and recommended. However, tags may be weighted by users who have the same taste. The similarity among weighted lists of tags should be considered, which is not discussed in [27]. For weighted tags, we use equation (9) to describe the similarity.

$$SemanticSim(X, Y) = \sum_{x_i, y_j \in SST}\left(\frac{W_{x_i}}{W_{x_{all}}} * \frac{W_{y_i}}{W_{y_{all}}}\right) \quad (9)$$

where SST is the set of semantic similar tags, $W_{x_{all}}$ $W_{y_{all}}$ is the sum of weight in X and Y. The result should not exceed 1. Thus can be used as a similarity score directly.

D. Similarity merge

Our method considers that the similarity of movie ontology is given by the similarity of distinctive features. It's hard to say which feature has the most importance on the interesting of users. Here we used the candidate set consisted of the movies with the maximum similarity score for each feature, and the average rate score by user i used as the estimate rate value of the movie for this user. Equation (10) shows the process.

$$\hat{r}_{u_{ij}} = avg\{r_{u_{i\hat{j}_m}} | \hat{j}_m = maxSimScore \in feature_m\} \quad (10)$$

where $\hat{j}_m$ is the index of movie with maximum similarity score for feature m. Furthermore, if the user has no rate score for the whole similar movie set, the estimate rate score will be zero. It has no impose on the original collaborative filtering method.

Fig. 3 shows the whole process of proposed recommendation. The ontology-based similarity estimate processing is showed in Algorithm 1. Firstly we get the ontology structure of movies, and the get the similarity candidate set for each feature from line 2 to 7. From line 8 to 11 we find the most similar movie for each feature. Estimating rate score is based on the average score of movies in candidate set that have already rated by user in line 12. Then the original collaborative filtering will be performed with the new filling user-item matrix to get the recommended list. After CF process, the rough recommended list will be clustered based on the ontology similarity and improve the diversity of recommender system, as for optional step.

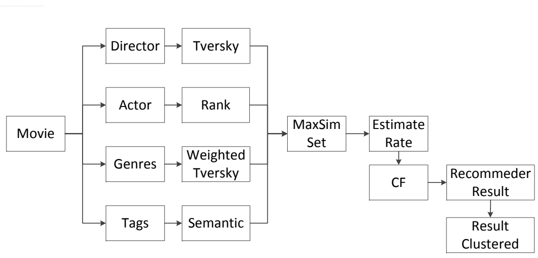


Figure 3. Flow chart for the proposed recommend process.

TABLE I.
HETREC'11 DATASET

Parameter	Count
movies	10197
users	2113
genres assignments	20809
directors	4060
actors	95321
tags assignments	47957

V. PERFORMANCE EVALUATION

This section describes the experimental results to validate our proposed method using the HetRec 2011 MovieLens dataset published by GroupLens research group [28]. The dataset is an extension of MovieLens dataset that is linked to the Internet Movie Database (IMDb) [29] and Rotten Tomatoes Movie review systems [30]. Table I shows the details of dataset.

A. Experiment setup

The testing method is similar with the one in [25]. The whole ratings are sorted by timestamps and we could get a probe set (latest 10%) and the training set (90%). Four

Algorithm 1 Ontology-based Similarity Estimate Processing

Require:

D_i = the Directors of the movie;
A_i = the rank for actors of the movie;
G_i = the weight of genres in the movie;
T_i = the weight of tags in the movie;
$U_{watched}$ = the watched movies of the user;

Ensure:

Estimate rate value of the movie set for this user $\{\hat{r}_{u_i}\}$

```
for movie i unwatched by user u do
    for movie j watched by user u do
        D_ij ← TSim(D_i, D_j)
        A_ij ← RankSim(A_i, A_j)
        G_ij ← TWSim(G_i, G_j)
        T_ij ← SemanticSim(T_i, T_j)
    end for
    j_D ← argmax_{j∈U_watched} D_ij
    j_A ← argmax_{j∈U_watched} A_ij
    j_G ← argmax_{j∈U_watched} G_ij
    j_T ← argmax_{j∈U_watched} T_ij
    r̂_{u_i} = avg{r_{u_{j_D}} + r_{u_{j_A}} + r_{u_{j_G}} + r_{u_{j_T}}}
    {r̂_{u_i}} += r̂_{u_i}
end for
return {r̂_{u_i}}
```

separate test sets are created to test proposed method. T_1 contains old movies that have been rated by old users who have rated more than one movie in a training set. T_2 contains new movies that have never been rated in the training set. T_3 contains randomly selected ratings, and T_3' is the long tail part of T_3. Additionally, a 5-fold cross validation test (80% training set, 20% probe set) is performed utilizing the optimized parameters of each method as Table II shows. We randomly select 1000 additional movies unrated by user i, and predict the rating for the test movie and other 1000 movies. p is rank of the test movie within the recommend list. A top-N recommendation list is picked by the top N ranked movies. $p \leq N$ means hits with the value 1. Otherwise the value is 0. The overall recall is defined over all test cases, as equation (11) shows.

$$recall(N) = \frac{\#hits}{|T|} \tag{11}$$

B. Software implementation

The runtime is reported based on MATLAB codes running on a machine with an Intel 3.1-GHz CPU and 8 GB RAM. Tversky and Rank Similarity function could be realized directly from equation (5) and (7), and LDA MATLAB toolbox could be used to group tags by the semantic similarity. The estimation process is an independent part of the whole recommendation. There is no need to modify collaboration filtering. The filling user-item matrix will serve as the input of CF algorithm, instead of original one.

TABLE II.
PARAMETERS USED FOR THE ALGORITHMS ON THE HETREC'11

Algorithm	Parameter	Hetrec'11
NNCF	neighborhood size	25
PureSVD	latent size	50
LDA in Ontology	topic count	50

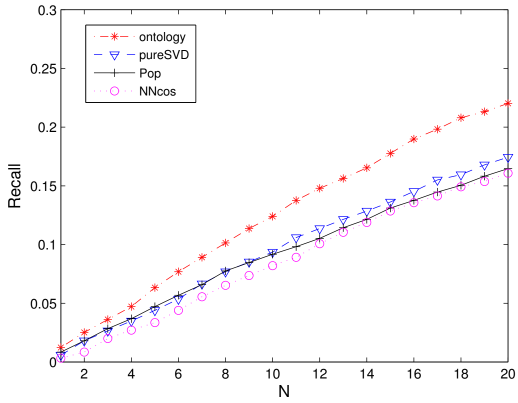


Figure 4. Recall (k) (training set Mts, test set T_1).

C. Comparing with baseline methods

In our test, we have first analyzed the baseline algorithms described in Section III. (1) TopPop. (2) NNCF (non-normalization user-based CF with cosine similarity). (3) PureSVD (CF based on latent-features).

Figure 4 shows the performance of ontology-based method and other baseline methods. PureSVD shows the best result in the baseline methods, and the ontology-based method got even better. However, owing to the limited size of test data, it still hard to fully show the performance of CF algorithms (NNCF got the same result with TopPop). Figure 5 shows the consequence in terms of $recall(k)$ and obtained by performing 5-fold cross validation test. PureSVD still got the best in baseline methods while the next is NNCF method. Onotology-based methods got the best result.

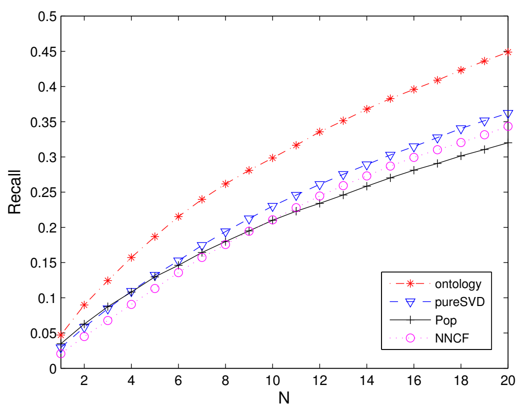


Figure 5. Recall (k) (training set Mts, test set $5cross$).

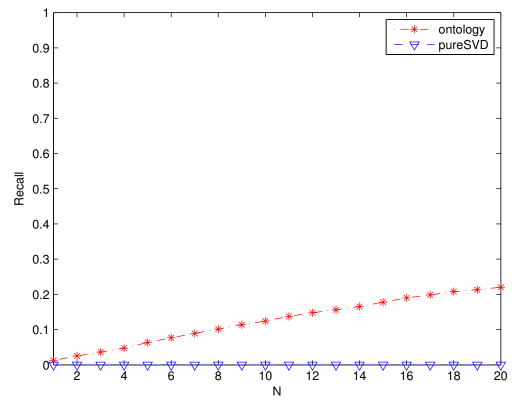


Figure 6. Recall (k) (training set Mts, test set T_2).

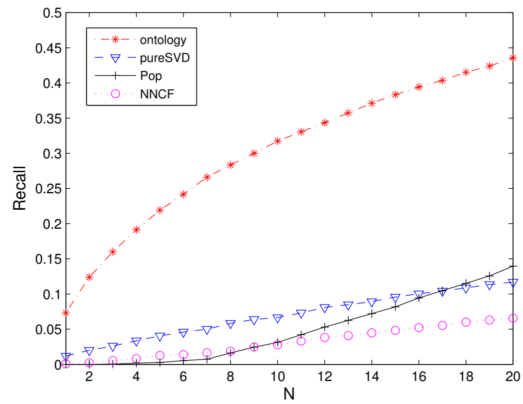


Figure 8. Recall (k) (training set Mts, test set T_3').

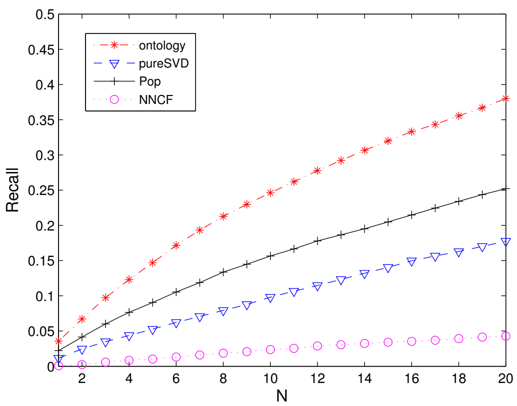
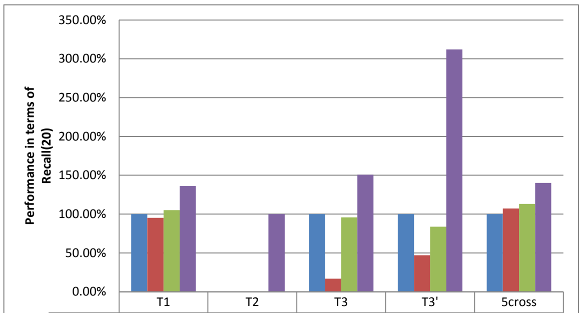


Figure 7. Recall (k) (training set Mts, test set T_3).



	T1	T2	T3	T3'	Scross
TopPop	1	0	1	1	1
NNCF	0.9516	0	0.1694	0.4704	1.0731
PureSVD	1.0503	0	0.9593	0.838	1.1313
Ontology	1.3616	1	1.5081	3.1215	1.4023

Figure 9. Relative performance comparison in terms of recall (20).

D. New Items & Long Tail

Figure 6 shows the new items problem of original pureSVD. As mentioned above, traditional collaborative filtering algorithms such as NNCF and PureSVD suffer from the new item problem. The newly added items could not be recommended until they got sufficient rate score by users. While ontology-based method could find the inherent relationship between movies. The recommendation for the most similar one is based on innate similarity.

Figure 7 shows the results using the test set T_3 which contains randomly sampled 5-star ratings from the probe set. PureSVD performs not well because of the new items problems. TopPop method performs even better in the average case. Ontology-based method still performs well in this case.

Figure 8 shows the long tail recommended result. In this case, the TopPop method decreased and shows lowest performance, while PureSVD performs better in Top-10 recommender result.

Figure 9 summarizes the experiments result and shows the relative best-case performance of each baseline method in terms of recall(20) compared to TopPop for different test sets. We can find that ontology-based method performs well for new items problem and long tail recommendation.

VI. Conclusions

In this paper, we propose ontology-based similarity to estimate the missing value in user-item matrix. Ontological features indicate inherent relations among items. The similarity between items is dependent on their properties, that is, common features tend to increase similarity and non-common ones tend to diminish it. Experiments using Hetrec'11 dataset is revealed the effectiveness of the proposed method compared with state-of-art methods.

Acknowledgment

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper. This work was supported in part by National Key Technologies R&D Program of China (GrantNo.2011AA01A107) and the "Strategic Priority Research Program" of the Chinese Academy of Sciences(Grant No. XDA06030900)

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[2] G. Karypis, "Evaluation of item-based top-n recommendation algorithms," in *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 2001, pp. 247–254.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[4] K. Zhou, S.-H. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 315–324.

[5] S.-T. Park and W. Chu, "Pairwise preference regression for cold-start recommendation," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 21–28.

[6] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.

[7] A. Tversky, "Features of similarity." *Psychological review*, vol. 84, no. 4, p. 327, 1977.

[8] I. Cantador, P. Brusilovsky, and T. Kuflik, "2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011)," in *Proceedings of the 5th ACM conference on Recommender systems*, ser. RecSys 2011. New York, NY, USA: ACM, 2011.

[9] K. Seyerlehner, A. Flexer, and G. Widmer, "On the limitations of browsing top-n recommender systems," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 321–324.

[10] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems (TOIS)*, vol. 23, no. 1, pp. 103–145, 2005.

[11] M. A. Domingues, A. M. Jorge, and C. Soares, "Using contextual information as virtual items on top-n recommender systems," *arXiv preprint arXiv:1111.2948*, 2011.

[12] L. Baltrunas and F. Ricci, "Context-based splitting of item ratings in collaborative filtering," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 245–248.

[13] P. Cremonesi, P. Garza, E. Quintarelli, and R. Turrin, "Top-n recommendations on unpopular items with contextual knowledge," in *2011 Workshop on Context-aware Recommender Systems. Chicago*, 2011.

[14] Y. Chen, C. Wu, M. Xie, and X. Guo, "Solving the sparsity problem in recommender systems using association retrieval." *Journal of computers*, vol. 6, no. 9, 2011.

[15] S. Gong, "A personalized recommendation algorithm on integration of item semantic similarity and item rating similarity." *Journal of Computers*, vol. 6, no. 5, 2011.

[16] X. Cui, G. Yin, L. Zhang, and Y. Kang, "Method of collaborative filtering based on uncertain user interests cluster." *Journal of Computers*, vol. 8, no. 1, 2013.

[17] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[18] B. Mobasher, X. Jin, and Y. Zhou, "Semantically enhanced collaborative filtering on the web," in *Web Mining: From Web to Semantic Web*. Springer, 2004, pp. 57–76.

[19] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2002, pp. 253–260.

[20] P. Cremonesi, R. Turrin, and F. Airoldi, "Hybrid algorithms for recommending new items," in *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*. ACM, 2011, pp. 33–40.

[21] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, "Ontological user profiling in recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 54–88, 2004.

[22] M. S. Aktas, M. Pierce, G. C. Fox, and D. Leake, "A web based conversational case-based recommender system for ontology aided metadata discovery," in *Proceedings of the 5th IEEE/ACM international workshop on grid computing*. IEEE Computer Society, 2004, pp. 69–75.

[23] I. Cantador, A. Bellogín, and P. Castells, "Ontology-based personalised and context-aware recommendations of news items," in *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, vol. 1. IEEE, 2008, pp. 562–565.

[24] D. Sánchez, M. Batet, D. Isern, and A. Valls, "Ontology-based semantic similarity: A new feature-based approach," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7718–7728, 2012.

[25] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 39–46.

[26] R. Bambini, P. Cremonesi, and R. Turrin, "A recommender system for an iptv service provider: a real large-scale production environment," in *Recommender Systems Handbook*. Springer, 2011, pp. 299–331.

[27] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 61–68.

[28] http://www.grouplens.org/, 2013, [Online; accessed 1-Aug-2013].

[29] http://www.imdb.com/, 2013, [Online; accessed 1-Aug-2013].

[30] http://www.rottentomatoes.com/, 2013, [Online; accessed 1-Aug-2013].

Haomin Cui received his B.S. degree in automation from Hefei University of Technology, Hefei, China in June 2009. He is currently working towards his Ph.D. degree at University of Science and Technology of China. His current research interest includes pattern recognition, data mining and recommender system.

Ming Zhu is professor in the Automation Department at the University of Science and Technology of China (USTC). His research interests are generally in the field of data mining, mass multimedia retrieval and pattern recognition.

Shijia Yao received the B.S. degree in computer science and technology and Circuit Heilongjiang University, Haerbin, China, in June 2007. He is currently working towards his Ph.D. degree in network communication system and control at University of Science and Technology of China, Hefei , China in 2008. His current research interest includes content delivery network and distribution cache system.