Volume-5, Issue-9, September - 2016 • ISSN No 2277 - 8160

Original Research Paper | Computer Science

# Personalizing Movie Recommendation Using Semantic Contents in Collaborative Filtering

**Kruti Jani** Asst. Prof. SCPPGICA, Ahmedabad, Research Student (1627), HNGU, Patan

**Dr. V.M. Chavda** Principal, NPCCSM, Kadi

**ABSTRACT** *Next generation recommender systems use Semantic Web technology and limit the problems of traditional approaches through inferences. In this paper, features extracted from Link open data cloud used to weight item-based collaborative filtering. This work deal with a large dataset, identify neighbors and predict the rating based on semantically weighted content. Evaluation using Open MovieLens datasets shows item based method and improved result than user-based collaborative-filtering*

**KEYWORDS :(Recommender System, Link Open Data, Semantic Web, Collaborative Filtering)**

## 1. Introduction

World Wide Web has entered into the era of Information Overload due to emergence of social networks and the publication of a massive amount of information on the Web: more information is produced than what we can really consume and process. The Web is moving from a Web of documents to a Web of Data. The Semantic Web technology and Linked Open Data (LOD) [1] initiative published a vast amount of RDF data. More and more semantic data enable to set up links between objects in different data sources by connecting information in a single global data space. This RDF can be feed to personalized information access tools such as Recommender systems [2]. RS enhance user experience using user choices, rankings, scores, tags, etc as input data, and adapt as form of ordered lists or scores of items to an output. Recommender systems using Semantic Web technology limits the problems in traditional RS by completing the incomplete information through inferences. In building recommendation systems, collaborative Filtering is one of the most promising recommendation techniques among the entire existing techniques. This method gather user's preferences for items, look for a set of neighbors sharing similar preferences, and infers rating on a particular item based on the information collected from neighbors. With the predicted rating, the system will recommend products which have high predicted ratings to users. This paper builds and evaluates semantics-aware collaborative filtering movie recommendation system using all the notions and elements of Linked Open Data.

## 2. Related Work

Collaborative filtering recommender system is a well studied topic. [2]. LikeMinds [3] is one of the most famous applications using collaborative filtering. In [4] SemanticSVD++, is presented incorporates semantic categories of items into the model. In [5] proposed collaborative method for integration of FOAF information with item information about DBpedia Ontology. In [6] semantically enhanced CF proposed where structured semantic knowledge about items is used in conjunction with user-item ratings to create a combined similarity measure for item comparisons.

## 3. Collaborative Filtering

Collaborative filtering systems [4] finds user and item based Nearest Neighborhood algorithms (like k-NN) as core practice. This section illustrates the concept and operation methods of item-oriented as well as user-oriented design. The following example in provides better understanding of the kNN algorithm.

**Table 3.1 Example Rating Matrix**

| | Movie1 | Movie2 | Movie3 | Movie4 |
|---|---|---|---|---|
| User1 | 3 | - | 4 | 2 |
| User2 | 4 | 5 | - | 4 |
| User3 | - | 4 | 2 | 3 |
| User4 | 3 | 5 | 2 | - |

All users in Table 3.1 have rated in a scale 1 to 5 for four movies and – represent the rating detail is not available for that movie by the corresponding user. Similarly each movie obtains ratings from four different users.

### 3.1 User-Based Method

The principle of the user-oriented nearest neighbor algorithm is listen to people that have similar thoughts [3]. In Table 3.2 shows the complete user-user distance matrix, which was calculated based on the cosine-similarity of the user/row vectors.

$$(\mathrm{sim}[\mathrm{vecA}, \mathrm{vecB}] = \frac{vec\ A\ .vecB}{|\mathrm{vecA}|\,.|\mathrm{vecB}|}) \qquad (1)$$

**Table 3.2 User- User Similarity Matrix**

| | U1 | U2 | U3 | U4 |
|---|---|---|---|---|
| U1 | - | 0.7389 | 0.6301 | 0.7006 |
| U2 | 0.7389 | - | 0.4523 | 0.7314 |
| U3 | 0.6301 | 0.4523 | - | 0.5307 |
| U4 | 0.7006 | 0.7314 | 0.5307 | - |

As the user similarity matrix in Table 3.2 is symmetric, we are able to safe computational time by only calculating the upper triangular of the similarity matrix. In addition, To get the final prediction for movie compute the mean of all user ratings on this movie. Note that all ratings are first multiplied with their according user similarity value before they are summed up.

### Item-based Method

User-oriented collaborative filtering suffers from scalability issue when no of users grow. An item-oriented collaborative filtering approach in [4] shows better scalability and improved prediction accuracy than ordinary user-oriented methods. Item oriented methods make predictions based on similarities between the rating patterns of items. In general, item-oriented methods are more reasonable, because users are more familiar with items previously preferred than with potentially like-minded users. In the following illustrate the algorithm of item-oriented rating prediction [3].

**Table 3.3 Item-Item Similarity Matrix**

| | M1 | M2 | M3 | M4 |
|---|---|---|---|---|
| M1 | - | 0.7389 | 0.6301 | 0.7006 |
| M2 | 0.7389 | - | 0.4523 | 0.7314 |
| M3 | 0.6301 | 0.4523 | - | 0.5307 |
| M4 | 0.7006 | 0.7314 | 0.5307 | - |

### 3.3 Semantic contents in collaborative filtering

Semantics weight the items and improves the prediction accuracy. For the classification, the movie features retrieved from the LOD cloud are represented in a feature vector. The proposed approach uses semantics of the content for filtering and transforms the RDF data in the feature vector. Some content-based method [2] proposed to allocate weights using statistical data of the items and scores to the items, while this approach use the content of items in order to determine the similarities of them and weight them. This additional semantic data weight more accurately the items by content-based method and subsequently enhancing the accuracy of prediction.

This proposed method follows three different stages [4]:

**Stage 1 - Retrieve and Pre-processing RDF data**
In order to use the information related to the movie, a query based on the movie label was designed in SPARQL language and by Post URL method is sent to the LOD Cloud.

**Stage 2 - Using item features to allocate weights to items (Weighting the items)**
Stage 1 gathered information about the genres, directors and actors of movies in dataset. This movie profile is compared to target movie profile based on similarity the weight of each movie is determined. Cosine-based similarity is used to measure the movie similarity here. In following example, suppose movies M and T has features { gi, di, ai } where gi denotes i-th genre, di denotes i-th director and ai denotes i-th actor in a movie respectively.

M = {g1, g2, d1, d2, a1, a2, a3}

T = {g1, g2, g3, d3, a2, a3}

The conjunction of the feature set of these two movies is:

M U T = { g1, g2, g3, d1, d2, d3, a1, a2, a3}

Thus, following is a constructed vector for each movie:

M = {1, 1, 0, 1, 1, 0, 1, 1, 1}

T = {1, 1, 1, 0, 0, 1, 0, 1, 1}

According to the pre-processing stage mentioned above the number of actors extracted for each movie is different. Thus, when any movie compared with the target movies, common actors between them are just reflected in M U T. accordingly in above example, actor al will be deleted from M and T vectors:

M U T= {g1, g2, g3, d1, d2, d3, a2, a3}

M = {1, 1, 0, 1, 1, 0, 1, 1}

T = {1, 1, 1, 0, 0, 1, 1, 1}

By this way the movie's profile is managed in the form of vectors, the similarity between two vectors can be measured using cosine-based similarity in the following way:

$$w_{MT} = \cos(\vec{M},\vec{T}) = \frac{MT}{\|M\|_2 \times \|T\|_2} = \frac{\sum M_i T_i}{\sqrt{\sum M_i^2}\sqrt{\sum T_i^2}} \qquad (2)$$

Mi denotes i-th element of vector M and Ti denotes i-th element of vector T. However, the result of this equation is generally between a value 0 and 1. Where 1 represents perfect similarity and 0 represents perfect dissimilarity. The elements are of two compared vectors are 0 and 1. The number of common ones (1s) calculated as per above equation is equal to the numbers of common features concerning two movies. Thus, if the weight is zero then considered such target item has not any common feature compared item.

**Stage 3 - Using of weight of items for neighbor selection**
As the previous stage weights the item, each item's weight is based on its similarity to the target item. Collaborative filtering methods first find the most similar users to active user and consider them as a set of the neighbors of active user and finally predict and provide suggestions. The goal of the prediction of target item's score is for Active User. The members who voted for target item only have been studied In order to create the set of neighbors. Standard Pearson correlation Criteria is used to evaluate the dependencies between scoring patterns of users and active user.

$$PC(a,u) = \frac{\sum_i (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_i (r_{a,i} - \bar{r}_a)^2 (r_{u,i} - \bar{r}_u)^2}} \qquad (3)$$

In the above equations users a and u have attributed to the items and and are the averages of total scores. The scores that users a and u have attributed to the i-th item respectively are in $r_{a,i}$ and $r_{u,i}$ [3].

### 3.4 Computing Item Similarities
The following equation 7 is used to compute the similarity of two users.

$$sim(a,u) = wpc(a,u,j).CF$$

$$CF = \frac{x}{50} \; if > 50 \qquad (4)$$

When the similarity of all users with an active user is found then set of neighbors are chosen for further process. Therefore according to the users' resemblance, users are chosen arranged in descending order. [4].

### 3.5 Rating Prediction
The set of active user's neighbors is the result of the previous stage. At this stage, by the use of the scores allocated to target item by set of the neighbors, the score of target item can be predicted. For this purpose, the equation 8, which is typically used in user-based collaborative filtering [2,4] is applied.

$$\overline{r}_{a,t} = \overline{r}_a + \frac{\sum_{u\in N_{i(a)}} (r_{u,t} - \overline{r}_u)\cdot sim(a,u)}{\sum_{u\in N_{i(a)}} sim(a,u)} \qquad (5)$$

$N_{i(a)}$ is set of active user's neighbors. and are the average of total scores that the users a and u have attributed to the items respectively[4].

## 4. Evaluation
In order to evaluate LOD-based RSs Movielens dataset is based on the released by the GroupLens research group is used. These datasets contain mappings between items (movies, artists, books) and their corresponding DBpedia URIs [5].

A Root Mean Squared Error calculation is used to determine the error in the prediction (p) compared to the real rating (r) after n predictions:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p-r)^2} \qquad (6)$$

The Figure 4.1 reveals that item-oriented approach performs much better than the user-oriented one as the semantic features incorporated.

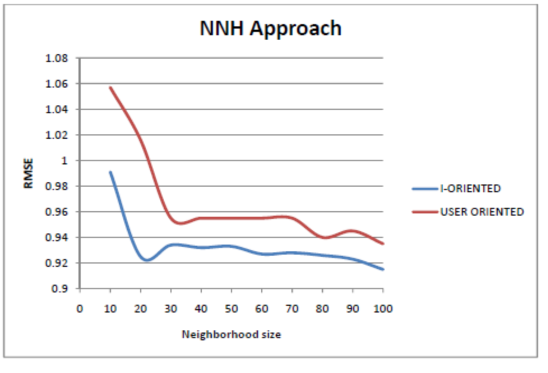


**Figure 4.1: Item-Oriented and User-Oriented NNH Approach**

Volume-5, Issue-9, September - 2016 • ISSN No 2277 - 8160

## 5. Conclusion

This paper presents semantically enhanced movie recommendation using collaborative filtering using LOD. Due to the multiplicity of content features extracted from the LinkedMDB and MovieLens data sources, there exist many possible item-user feature combinations that contain valuable information. This knowledge can be employed to design an advanced recommendation system with the ability to generate more individual and accurate prediction results.

## 6. References

[1] Lorna M. Campbell and Sheila MacNeill. The semantic web, linked and open data. In JISC CETI S. 2010.

[2] Gediminas Adomavicius and Alexander Tuzhilin "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, Vol 17, No 6, June 2005 [3] Jonathan L. Herlocker and et al "An Algorithmic Framework for Performing Collaborative Filtering".

[4] M. Rowe. Semanticsvd++: incorporating semantic taste evolution for predicting ratings. In 2014 IEEE/WIC/ACM International Conferences on Web Intelligence,WI 2014, 2014.

[5] B. Heitmann and C. Hayes, "Using linked data to build open, collaborative recommender systems" In AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, 2010.

[6] B. Mobasher, X. et al. "Web Mining: From Web to Semantic Web", volume 3209 of Lecture Notes in Computer Science, pages 57{76. Springer Berlin Heidelberg, 2004.