SemTopMF
Prediction Recomendation by Semantic Topics Through Matrix Factorization Approach

Nidhi Kushwaha and O. P. Vyas
Department of Information Technology, Software Engg. Lab., Indian Institite of Information Technology, Allahabad, India

Keywords: Matrix Factorization, Semantic Topics, DBpedia, RDF, SPARQL, Similarity Coefficient, TF-IDF.

Abstract: The Matrix Factorization model proved as a state of art technique in the field of Recommender Systems. The latent factors in these techniques are mathematically derived factors that are useful in terms of dimensionality reduction and sparsity removal. In this paper, we exploited the information on these latent factors in addition with semantic knowledge fetched from the DBpedia dataset to predict the movies to users, based on their selected topics in the past. We incorporate matrix factorization with the Semantic information to increase the accuracy of the recommendation and also increase the contextual information into it. For handling cold start users, we also provide an opportunity for the user, to select topics at the run time and prediction will be made according to their selection. To improve the diversity of the prediction in both the cases we also used a specific strategy for the end user recommendation.

1 INTRODUCTION

Recommender systems are not new, it has been in progress since 90's. Tapestry (User-Based CF) (Goldberg, 1992), GroupLens (Collaborative Filtering) (Resnick, 1994), NewsWeeder (Content Based) (Lang, 1995), Bellcore video recommender (Collaborative Filtering) (Hill W., 1995), InfoFinder (Content Based) (Krulwich, 1996), PHOAKS (Collaborative Filtering, 1997) (Terveen, 1997), Fab (Hybrid Based, 1997) (Balabanovic, 1997), PTV (Hybrid Based, 2001) (Cotter, 2001), Amazon (Item based Collaborative Filtering) (Linden, 2003), Pandora (Content Based) and plethora of applications are designed in this journey of research. RSs act as the agent (software) to recommend a person before s(he) explicitly shows his/her likings towards a particular item. It also helps a user to overcome the problem of knowledge overflow. Recommender Systems provide filtered information so that it help users to get the information according to their past preferences and also their similar group members (Bobadilla, 2013). Typical Recommender Systems consist of three basic entities: Users, Items and their ratings, either implicitly or explicitly (Adomavicius, 2005). Matrix factorization approach in the past, mainly known for the collaborative filtering approaches which deal with only rating matrix. It basically removes the dimensionality and the sparsity problem of item rating matrix (Zhang, 2006) (Rossetti, 2013) (Koren, 2009). In this case the input of the system is usually are the user ratings on movies that the user has already seen and the output of the system is the prediction of movies that a user would like in future. The prediction of the unknown ratings is based on patterns of partially observed rating matrix (in case of low rank rating prediction). This method differs from previously applied approach that also utilize content information about the user's (e.g. Age, gender explicitly given preferences) and item's in case of a movie (genre, year, actors, text reviews). Content based methods totally focused on features of the items and users, thus suffer from over-generalization and also lack of personalization. Collaborative method of recommendation, act as a complement of the Content based method, it needs only user item preference matrix, and give more importance of user-user collaboration. This approach also suffered from some drawback for ex: cold start user & cold start item problems. In literature, the problem has been removed through two ways, first from the data available online in the form of social network and another way is by using the combination of both Content based and Collaborative Recommendation

Kushwaha N. and Vyas O..
SemTopMF - Prediction Recomendation by Semantic Topics Through Matrix Factorization Approach.
DOI: 10.5220/0005475001180123
In *Proceedings of the 11th International Conference on Web Information Systems and Technologies* (WEBIST-2015), pages 118-123
ISBN: 978-989-758-106-9

System methods, also known as Hybrid methods for the recommendation (Burke, 2007) (Gemmell, 2009). In all the above three types Recommender Systems, namely Content Based, Collaborative and Hybrid two methods are used for predicting ratings, i.e. Memory Based and Model Based approaches depending on the utilizations of memory.

Resultant of Recommender System also divided into two parts, predicting the rating and determining the rank of the predicted rating, respectively. The bottleneck for Memory based recommendation is space and processing time of the whole data set, while in Model based the main problem is complex and time consuming of running the algorithm. The performance of RSs degrades with the increase of the number of item's and number of users. Despite the challenges of over-generalization, the cold start Recommender System also suffered from high dimensionality and sparsity (Balabanovic, 1997) (Adomavicius, 2005) (Rossetti, 2013).

In case of linear factor model for n users and m items, the rating preferences with respect to k- factor model are given by the product of a nxk , whose column represent factors of user's and a kxm factor matrix Z' whose rows represent the factors of items. Thus, in this way a linear factor model is obtained by approximating the observed rating preferences Y with a low-rank matrix X. This low rank matrix X should be obtained from the minimization of Root Mean Squared error to obtain an original matrix Y. It is difficult to find out global minima; because of original sparse matrix Y. Hofmann in 2004 proposed Loss function in place of Root Means squared Error. However the idea becomes very popular with the variation of matrix factorization approaches (Adomavicius, 2005) (Zhang, 2006) (Rossetti, 2013) (Koren, 2009) but it always suffers from the lack of human interpretation. In this paper, authors exploit the features retrieved from the Semantic Web (SW) (Bizer, 2009) with the combination of mathematically generated information from matrix factorization to make it more meaningful and valuable. Web3.0 develop an environment through which we can share the information in machine readable format and in the unified way (Bizer, 2009) (MacNeill, 2010). This information grows day by day that encourage researchers to utilize this information for the cutting edge applications like Data mining, Human Computer Interaction, Information Retrieval and Recommender Systems. The concept of Semantic Web was initiated by Sir Tim Berners Lee that formed big project named Linked Open Data project (Bizer, 2009). Connecting data with the related information is the main aim of this project. For this task various researchers came forward to give their contribution in the standardized format, i.e. in Resource Description Framework. The idea of keeping this data open, benefited others by linking their organization's specific content and thus increases its accessibility to all. Data associated with the particular entity in the Semantic Web can be fetched with SPARQL querying (Prud'hommeaux, 2008) (Broekstra, 2012) on the stored RDF (Resource Description Format) storage.

In the related work, the authors first highlighted the state-of-art techniques of RS and its characteristics (without SW) in Section 2. Authors also highlight the proposed a model in section 3. At the end, the paper summarizes with a conclusion and future work with Section 4.

2 RELATED WORK

In Recommender System there is a set of user, items and the ratings provided for these items are given as input. The output should be the ratings for each user to the items which was unknown previously. In the R_u matrix the rates are provided by each user that belongs to [1...5], without the loss of generality, we map the interval of ratings into [0,1].In Semantic Web graph information related to items and their associated characteristics are already present using standard XML like language called as RDF (Resource Description Framework), note that the links are unidirectional. To utilize this information in a meaningful way it is necessary to calculate the weight of each feature which denotes the importance over all movies features. Combining the information of contents generated from Semantic Web with benchmark dataset's *Ru* matrix is the main motivation of this work.

As discussed earlier Collaborative Filtering methods of Recommender Systems have been used in two different ways one for neighbourhood methods and other for Latent factor models. In our paper we choose Latent factor models as they can work efficiently on the small datasets thus efficiently solve the scalability issues as well as computational time complexity. The method of Latent Factor model, also known as Matrix factorization method, it maps both users and items into a joint latent factor space with the dimensions f, so that the inner product of that space can be modelled as interaction of user-item cell. Suppose after factorization the vector associated with a user is $u_f \in \mathbb{R}^f$ and the vector that associated with an item is $i_f \in \mathbb{R}^f$. For a given item i, the element of i_f denotes the importance of

feature in the form of weighted factors, either positive or negative. Similarly, for a given user *u,* the element of u_f shows the measure up to which extent a user has liked features of items, again, either positive or negative. The resultant dot product $u_f X$ $i_f{}^T$, should represent the original rating matrix, where each cell denotes the ratings of users for items approximately, as shown with Eq. (1).

$$\sim R_{ui} = u_f * i_f{}^T \quad (1)$$

The above described model is very similar to an SVD model in which a small number of factors are chosen to predict the rating of the unknown items. This method mostly suffers from lots of missing ratings problem in the user-item rating matrix. Earlier Recommender Systems used Imputation method and Association Retrieval technology to overcome the problem of data sparsity that results in a dense matrix by computational expensive methods. Using Associative retrieval technology to explore the transitive associations based on the user's feedback data, realized a new collaborative filtering approach to alleviate the sparsity problem and improved the quality of the RS (Chen, 2011). It also has the possibility of data distortion in the original data. Hence, the solution proposed in (Liu, 2013) (Zhou, 2011) (Zhou T. S., 2012) (Kleeman, 2007) suggested regularization to overcome the problem of over fitting occur due to unbalanced data. The Eq. 2 explains the regularization function. Here, N represents the training set that consist of the set of the (u,i) pairs for which r_{ui} is known and constant λ controls regularization and also known as cross-validation.

$$\min_{u*,i*} \sum_{(u,i)\in N} (r_{ui} - u_f * i_f{}^T)^2 + \lambda(||u_f||^2 + ||i_f||^2) \quad (2)$$

There are many learning algorithms are proposed mainly ALS (Alternating Least Squares and Gradient Descent).Other than that for updating in the original formula, researchers also applied different biases for calculating the actual prediction for the users. These biases are mainly depending towards the behaviour of the user's liking and an item's popularity. Due to shortage of data provided by users in the form of ratings, some researchers add other biases that represent age, gender or implicit ratings given by each user. Other than that, biases also added that consist temporal aspects of users and movies. In recent years, Latent Semantic Analysis and pLSA (Kleeman, 2007) were proposed to originally develop the context of information retrieval systems. They both are dimensionality reduction techniques and also based on matrix decomposition similar to matrix factorization, SVD, PCA.

Probabilistic Matrix Factorization method (Salakhutdinov, 2008) perform well on very sparse and imbalanced dataset but a careful tuning is needed to avoid the over fitting. Bayesian Probabilistic MF (BPMF) overcomes this drawback by using Markov Chain Monte Carlo (MCMC) method that improves the accuracy of the RS. Bayesian Probabilistic MF with Social Relation (BPMFSR) also proposed by Tinghui et al. that assumes different hyper-parameters for the different users. The main drawback of the system is the uniform consideration of the item parameters that has been removed by Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents (BPMFSRIC) (Liu, 2013), the authors fuse item contents as well as social relations and item parameters are sampled according to the item contents. The main drawback of the method, was the use of trust information while ignoring the distrust information that had given in most of the online social networks. Also, the paper ignores the indirect trust information and only using or based on the direct trust relationship.

3 PROPOSED WORK

For the goal of a Recommender System is to generate meaningful recommendations to a collection of users for items or products that might interest them. Our goal is to explore ways towards attaching semantics to the latent factors of a matrix factorization model, such that (parts of) these models can be applied to new users (i.e. Users without or with only few known ratings) or can be exploited for explaining their recommendations. The work, therefore, constitutes a first step towards this direction by making the following contributions: 1) Acquisition of a dataset with unary ratings via a user study that can serve as ground truth for the development of portable and interpretable MF (Matrix Factorization) models. 2) Empirical results on identifying topic related factors and predicting topical interests of participants in our user study. In this paper, we apply the model on the MovieLens 1M datasets that consist of 1,000,209 ratings from 6,040 users and 3,706 different movies. Ratings are integer values from an ordinal scale ranging from 1 to 5, where 1 denotes worst and 5 represents the best feedback from the user (see Figure: 1, that shows the items rated by the users).

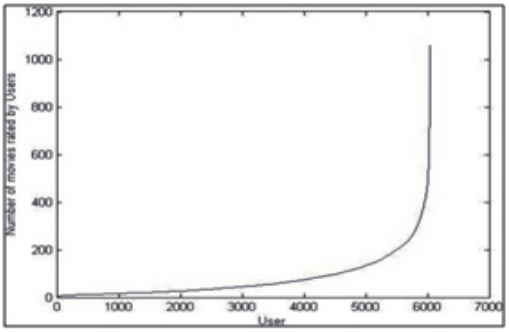

Figure 1: User Vs. number of items rated by User.

We are also using RDF dataset named DBpedia, for semantic feature extraction. This dataset contains 1,604 links with Movielens dataset thus provide an opportunity to extract the related features from it (as mentioned in Figure 2). The information about the 2-level categorical information of each movie is mentioned in the Table. 1.

Table 1: Explanation of DCtermTopics from DBpedia Movie graph.

DBpedia Movie Domain	1-Hop	2-Hop	3-Hop
http://dbpedia.org/page/Fled	http://dbpedia.org/resource/Category:English-language_films	http://dbpedia.org/resource/Category:Films_by_language	http://dbpedia.org/resource/Category:Works_by_language
http://dbpedia.org/page/Boomerang_1992_film	http://dbpedia.org/page/Category:American_romantic_comedy_films	http://dbpedia.org/page/Category:American_romance_films	http://dbpedia.org/page/Category:Romance_films_by_country

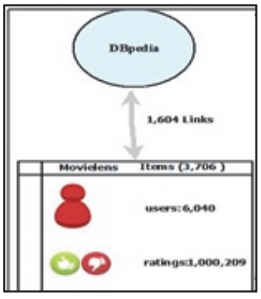

Figure 2: Links common between MovieLens & DBpedia.

The importance of each category corresponding to each movie has been explained by a weighting factor that depends on the Term and Inverse document frequency (TF-IDF) of that category in the relevant graph. Three main important questions are solved in this work are as follows:

- Given the latent factors from a factorized rating matrix, how can those latent factors merged efficiently with the content information that is retrieved from Semantic Web data.
- By merging the content, information it can improve the performance of the rating prediction or not.
- Will the system is able to provide appropriate ratings in the case of cold start users.

3.1 Method

In our approach we have used Nonnegative Matrix Factorization method to factorize the primary Rating matrix, i.e. User Item matrix *R*, where *R* represents a non-negative rating matrix where the rates are in binary form $R_{i,j} \in \{0,1\}$. The resultant of this factorization method produce two matrix Uf and If where R $\approx$ Uf*If^t. The Nonnegative factorization method gives a purely additive outcome in contrast to other dimensionality reduction techniques like Principal Component Analysis (PCA), see Figure: 3,

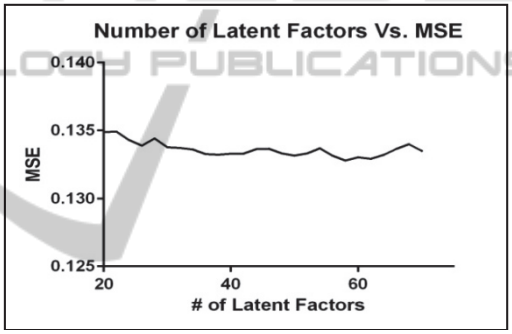

Figure 3: Number of Latent Factors Vs. MSE for 1M-Movielens dataset.

that shows error of factorization matrices while considering on different Number of Latent Factors. Eq. 3 shows this phenomena, N dimensional measurement vectors r^t (t=1,.....,T) $\in R^t$, a linear approximation of the data is given as,

$$r^t \approx \sum_{i=1}^{M} uf_i \; if_i^t = \text{Uf} * If^t \qquad (3)$$

With the factorize matrix, the inclusion of a content matrix, generated from the semantic data is also performed to work differently from the previous methods that uses matrix factorization. In this work we have combined the original rating matrix *R* with the content matrix $\mathbb{R}_{j,t}$, where *j* and *t* denotes the items and topics respectively. We have used weighted factor to fill the matrix instead of using only binary values that denote the presence of particular topic for an item in the matrix. We have used techniques described in FeGeLOD (Paulheim, 2012). Categories information of the particular URIs is explained by *dcterm*. For *dcterm*, TF-IDF is based

on the following formula:

$$\text{TF-IDF(Dcterm)} = (1/\text{count}) * \log N/|l_r(O_r)| \quad (4)$$

Here, the N = Total number of domain specific resources, like in our case total no. of movies present in the link data is 2979. $l_r(O_r)$ in Eq. (4) denotes, the number of other resources that have the same relation as target resource. This formula reduces the impact of most frequent relation by the multiply factor i.e. Count. Table 2 shows actual weights for a the RDF excerpt taken into consideration. Including content matrix, two original matrices namely rating (*U* x *I*) and content (*I* x *DctermTopic*) are used in combination to generate *U* x *DctermTopic*. One of the factored matrix, *Lf* x *M*, and *I* x *DctermTopic* matrix produce *Lf*x*DctermTopic* matrix, see Table.3 for the description of the Metrics.

Table 2: Top-10 weights for the Category: Golden_Bear_Winners.

Movie Name	Weights
Alphaville (1965)	0.639
Grand Canyon (1991)	0.384
Mangolia (1999)	0.338
Sense and Sensibility (1995)	0.303
In the name of Father (1993)	0.303
Thin Red Line (1998)	0.303
Cinderella (1950)	0.288
12 Angry Men (1957)	0.262
Rain Man (1988)	0.198
People vs. Larry Flynt (1996)	0.169

Table 3: Description of matrices.

Notation of Matrix	Matrix it denotes (Rows*Columns)
U*I	Users*Items
U*Lf	Users*Latent Factors
I*DctermTopic	Items *Topics (2-hop)
U*DctermTopic	Users*Topics
Lf* DctermTopic	Latent Factors*Topics
Lf*I	Latent Factors* Items
Lf*Count	LatentFactors*Frequency Count

After selection of topics from user the procedure goes as follows:

- After selecting the topics, it would return a set of Topics that indirectly returns Latent Factors hidden in that topic through the Lf x DctermTopic matrix.
- To obtain a set of LF that implicitly shows the importance of the each topic following work has been done:
  - Sort the matrix *Lf* x *DctermTopic* matrix (row-wise) in the descending order of the latent factor weights that present in the above matrix.
  - After, sorting the column vector of the selected topics, choose the top N_{lf} latent factor for further processing, and proceed it as follows: Maintain a matrix called *Lf*x*Count* of the dimension of (N_{lf}xT_{lf}) where N_{lf} denotes number of latent factor and T_{lf}=5 represent the highest importance of the latent factor and T_{lf}=1 denotes the least important,
  - This matrix created to obtain the set of common as well as important latent factors that explicitly represent the user preferred topics.
  - This matrix used to generate the count of which, each latent factor appear in the Topics preferred by the user. Also the count denotes the frequency of the same latent factors preferred by user implicitly in the form of preferred topic.
- Using this LF order, we sort the matrix LfxI, in which first row represents the Lf1 that is highly preferred by a user. Thus, this matrix is sorted so that it represents the more suitable movie first and the less suitable at the last. Based, on the sorted movies we recommend the user more suitable Top-N movies having similar factors as liked by him in the past. In the next step, the authors describe the inclusion of diversity after the step of Top-N recommendation.

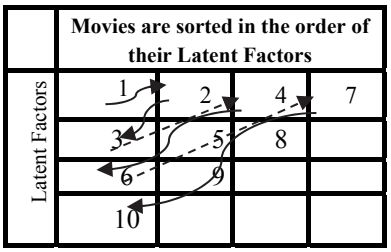

Figure 4: Diversity improvement in matrix Lf*I.

3.2 Diversity Improvement

Previous step is able to provide recommendations based on user's selected topics; however improvement of the diversity in the recommendation list is also needed because the prediction of the items solely depends on the content of the items. Figure: 4 shows the order of the recommended items. Dotted lines represent movement and dark lines shows recommendation of items that come in the way.

4 CONCLUSIONS & OUTLOOK

In this position paper authors used DBpedia topics for the Recommendation of users that also comprises with the matrix factorization approach and used traditional rating matrix. The blending of the topical information with the rating matrix is an important task of this The system used graph database and querying language to deal with it. We have proposed to develop a Recommender System to change adaptively according to the response of the user's selected preference. For the future authors are interested to also include social or trust based information in the system.

REFERENCES

Adomavicius, G. T. (2005). 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* .

Balabanovic, M. a. (1997). Fab: content-based, collaborative recommendation. *Special Section: Recommender Systems* , 66-73.

Bizer, C. H. (2009). Linked data – the story so far. *Int. J. Semantic Web Inf. Syst* .

Bobadilla, J. O. (2013). Recommender systems survey. *Knowledge-Based Systems* , 109-132.

Broekstra, J. K. (2012). Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. *ISWC*.

Burke, R. (2007). Hybrid Web Recommender Systems. *The Adaptive Web, LNCS* , 377-408.

Chen, Y. (2011). Solving the Sparsity Problem in Recommender Systems Using Associations Retrieval. *Journal of Computers*.

Cotter, P. S. (2001). PTV: Intelligent Personalized TV Guides. *Proceedings of the 17th National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence* (pp. 957-964). AAAI.

Gemmell, J. S. (2009). Hybrid Tag Recommendation for Social Annotation Systems. *CIKM*, (pp. 26-30).

Goldberg, D. N. (1992). Using Collaborative Filtering to weave an information TAPESTRY. *Communications of the ACM* .

Hill W., S. L. (1995). Recommending And Evaluating Choices In A Virtual Community Of Use. *CHI* .

Kleeman, A. H. (2007). Matrix Factorization for Collaborative Prediction. *ICME*.

Koren, Y. B. (2009). Matrix Factorization Techniques for Recommender Systems. *IEEE Computer Society*, (pp. 42-49).

Krulwich, B. B. (1996). *Learning user information interests through the extraction of semantically significant phrases.* AAAI Technical Report.

Lang, K. (1995). NewsWeeder: Learning to Filter Netnews. *ML* .

Linden, G. S. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE,* (pp. 76-80).

Liu, J. W. (2013). Bayesian Probabilistic Matrix Facorization with Social Relations and Item Contents for recommendation. *Decision Support system, Elsevier* .

MacNeill, L. M. (2010). The Semantic Web Linked and Open Data. *JISC* .

Paulheim, H. F. (2012). Unsupervised Feature Generation from Linked Open Data. *In: International Conference on Web Intelligence, Mining, and Semantics (WIMS'12)*. ACM.

Prud'hommeaux, E. S. (2008). *Sparql query language for RDF*. W3C .

Resnick, P. I. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *CSCW, ACM.*

Rossetti, M. S. (2013). Towards Explaining Latent Factors with Topic Models in Collaborative Recommender Systems. *In Proc. 24th International Workshop on Database and Expert Systems Applications (DEXA)* (pp. 162-167). IEEE.

Salakhutdinov, R. M. (2008). Probabilistic Matrix Factorization. *NIPS* .

Terveen, L. H. (1997). PHOAKS: a system for sharing recommendations. *Communications the ACM* , 59-61.

Zhang, S. W. (2006). Learning from Incomplete Ratings Using Non-Negative Matrix Factorization. *In Proc. of 6th SIAM Conference on Data Mining (SDM)*, (pp. 549-553).

Zhou, K. Y. (2011). Functional Matrix Factorization for Cold-Start Recommendation. *SIGIR.* ACM.

Zhou, T. S. (2012). Kernelized Probabilistic Matrix Factorization: Exploiting Graphs and Side Information. *SIAM International Conference on Data Mining (SDM).*