Inclusion of Semantic and Time variant information using Matrix Factorization Approach for Implicit Rating of Last.Fm dataset

Nidhi Kushwaha, Shubham Mehrotra, Ronish Kalia, Dhruv Kumar, and O P Vyas

Indian Institute of Information Technology, Jhalwa
211012 Allahabad, India
kushwaha.nidhi12@gmail.com

Abstract. Linked Open Data provides an opportunity to employ openly available metadata for the use of various applications. Recently it has been received great attention from the Recommender System community, to incorporate this semantic data as a side information for getting more accurate results. Despite the popularity of these Recommender Systems, it suffers from the problem of data sparsity and high dimensionality. Moreover, the utilization of semantic data becomes a bigger challenge, when user preferences are defined implicitly, instead of in a fixed rating scale. The above mentioned challenges necessitate us to introduce a comprehensive framework that is able to leverage the additional source of information for boosting the accuracy of the existing systems. In this paper, we have proposed a modified Joint Matrix Factorization approach for incorporating semantic information related to items and Tag based information with an implicit user preference for boosting the accuracy of the overall system. The model adheres to the phenomena of time variant Recommender System, thus it also utilizes the time related information of items. Experimental results show that our method gets more accurate recommendation results with faster converging speed than other existing matrix factorization based approaches.

1. Introduction

Linked Open Data Cloud Project was initiated by Sir Tim Berners Lee [3] [4] with the idea of assembling large volume of open and public data, that can be used and distributed freely across application boundaries. It is mostly represented in the form of Resource Description Framework (RDF), which is in a triplet structure i.e. subject, predicate and object. The RDF data model can be easily understood by human beings as well as by machines. The semantics of this background data enabled its utilization for building various applications. The Recommender System (RS) community also cannot afford to ignore the vastly available and rapidly updating knowledge base of Linked Open Data (LOD). Recommender Systems (RSs) help customers to overcome the problem of information overloading[13]. Broadly, there are three types of RSs: Collaborative, Content Based and Hybrid. Pure Collaborative filtering produces recommendation solely based on the user relationship with the items. However, pure Content Based systems depend on the contextual information related to the user profile and item features. The RSs which utilizes both the information is called as Hybrid System[13]. The quality of these RS is largely determined by the quality and quantity of the information used to build up the RS algorithms.

Recently, researchers has proven that adding the side information with the user-item ratings will improve the performance of the RSs [21]. More specifically, intelligently choosing and leveraging side information can be able to greatly affect the performance of the RSs. Semantically arranged data in the form of LOD can be utilized to exactly accomplish the above mentioned intuition of utilizing the side information for the recommendation. It potentially benefited the RS applications by providing closely related information of a particular item or product [8][25] [9][18].

Although, these RS leverage on well established technology & tools, new challenges arise when they will exploit the huge amount of semantic data and data sparsity in it. Furthermore, RS already suffered from the sparsity of user-item preference matrix due to the presence of less number of ratings. These challenges include high dimensional data and data sparsity. Other than that, existing LOD based RS approaches can handle only an explicit user preference, thus they are incapable to handle implicit user preference. Matrix Factorization (MF) and its variants are being used for RSs since 2009, and have successfully worked on both the above issues of sparsity and high dimensionality for generating reliable recommendations. Initially, MF approaches were adopted to predict unknown ratings only from the information present in user-item preference. Unfortunately, this approach proved to be inadequate for predicting recommendations, due to sparsity in the user-item preference matrix. The choice we makes is significantly influenced by one of the variants of MF approach, especially in the presence of implicit user-item preference matrix. The proposed work leverages modified Joint Martix Factorization (JMF) approach and exploiting implicit user preference, to handle all three above mentioned issues. The JMF approach was intitally proposed by Yue Shei et al. [21] in 2013 for incorporating non-semantic side information with explicit user-item preference matrix. However, our work significantly departs from the existing work by exploiting semantic features of item's & implicit user-item preference with modified JMF. The validation of the proposed work has been done with other well established algorithms, which shows its worthiness among all.

In the next section, the authors empirically investigate the basic MF approaches and its different variants, proposed specially for RS. This section also answers, how our proposed algorithm differs from the existing MF approaches. Section 3, introduces the preliminaries of the proposed work followed by proposed model in Section 4. Section 5 presents an insight of the implementation details and the extensive analysis of the proposed work. Lastly, Section 6 concludes our work and presents the future outlook.

2. Related Work

Approaches for RSs are conventionally divided into two methods, i.e. Memory Based and Model Based, depending on the approaches used for predicting recommendations. Although, Memory based methods are easy to understand, they are not scalable as they have to store the full matrix in memory. On the other hand, Model based approaches only store a model of the full matrix, and hence they are scalable. On the downside, Model based approaches are more complex to understand in comparison to Memory Based approaches [13]. Both the approaches are being exploited with implicit and explicit user preferences. This section discusses a Model Based approach i.e. MF and it's variants that was utilized for generating recommendations from explicit as well as implicit user preference. Conventional Matrix factorization (MF) methods provide an opportunity to express the explicit user preference into a known number of latent factors of the individual user as well as the item. These latent factors can be then utilized further for reproducing the user-item preference matrix. Nevertheless, these conventional MF approaches have been extended in recent past by many researchers to incorporate the side information rather than solely depending on the user-item preference matrix. Firstly, Korean et al. [14] introduced MF approaches for rate prediction using explicit user-item preference. It utilizes MF with various biases (user and item bias) and temporal dynamics for predicting recommendations. They concluded that MF method significantly improves the quality of RSs. Another direction of related work was proposed by Singh et al. [24] which instigated a recommendation model that simultaneously factorizes both, explicit user-item preference and side information of the item or user. This inclusion of side information in factorization model has been termed as the Joint Matrix Factorization (JMF). It serves to compensate the sparseness of the original user-item preference matrix. This method decomposes both, user-item preference & side information, and model the latent factors obtained from it, to reproduce the user-item preference matrix with a minimum error. Based on JMF, Hao Ma et al.[16] proposed probabilistic method to incorporate social trust information as side information with explicit user-item preference for improving the recommendations. Here, social trust was explicitly defined in the used dataset. This notion is further improved by Mohsen et al.[1] in 2010, by introducing trust propagation in both direct and indirect neighbors, rather than only depending on the direct neighbors, as done in the previous approach. Here, trust propagation was not given explicitly, but calculated using a mathematical formula. Tag based user-user similarity matrix was also exploited as a side information with the combination of user-item preference, by Zhen et al.[27] in 2008. The authors concluded that the incorporation results in better accuracy. Recently, Shi et al. [22] [23] explored mood-specific similarity and location similarity matrix of the users as the possible source of side information in 2010. More recently, Juntao et al. [15] proposed additional sources of information of users and items to model it with Bayesian Probabilistic MF in 2013. The authors especially considered direct trust information of the users and envisioned to incorporate distrust and indirect relationship between the users as future work. Aaron et al. in 2013 [17] proposed a deep learning technique to generate music, audio features and associate it with the music latent factor using MF approach. Although, different applications were utilized different variants of MF but the essence of all of them are identical, which is to incorporate the relative side information for the effective and efficient recommendation.

Our proposed work influenced by this by utilizing different side information with implicit user-item preference matrix for better recommendations. In this paper, we include the semantics as well as user given tag information of items as the side information. We make and analyze the results for the following contributions:

– Presenting and discussing a way to convert implicit preference into explicit preference.
– Utilizing the semantic information of the items retrieved from DBpedia, we fetched the data from SPARQL querying and prepared the item-property matrix.
– Incorporating time variant factor with the model by exploiting Tag information of items.

– Analyzing the performance of the aforementioned side information on different parameters.

The next section describes the preliminaries of the proposed work. It includes mathematical introduction of basic MF and JMF, followed by the short description of methodology used for semantic feature generation in subsection 3.3. Time aware RSs is discussed in subsection 3.4.

3. Preliminaries

In this section, authors, first introduce some preliminaries of the basic MF and JMF methods. They are used to incorporate the contextual information into the basic MF techniques as mentioned above. Extraction and preprocessing of the side information are briefly explained later in this section.

3.1. Recommendation using Probabilistic Matrix Factorization

Suppose, in user-item matrix, we have i users, j items, and the rating values are within the range of [0,5] or [0,9]. r_{ij} represents the rating of the user i given to the item j, and $U \in R^{i \times d}$ and $V \in R^{i \times d}$ be latent factors of users and latent factors of items respectively. The dimension of latent factors should be less than the original dimension of the matrix, i.e. $d << j$. Given this situation the conditional distribution probability over the observed ratings is:

$$\Pr(R|U, V, \sigma_R^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} [N(r_{i,j}|g(U_i^T V_j), \sigma_2^R)]^{I_{i,j}^R} \tag{1}$$

where $I_{i,j}^R$ is the indicator function, that is equal to 1 if the user have given the rating to the item otherwise 0. $N(r_{i,j}|\mu, \sigma^2)$ represents a probability distribution with mean μ and variance σ^2. The function $g(x)$ is the logistic function, $g(x)$=$\frac{1}{1+exp(-x)}$, that makes possible to bind the values of $g(U_i^T V_j)$ in the range of [0,1].The zero-mean spherical Gaussian priors on the user and item property vector:

$$\Pr(U|\sigma_U^2) = \prod_{i=1}^{m} N(U_i|0, \sigma_U^2) \tag{2}$$

$$\Pr(V|\sigma_V^2) = \prod_{j=1}^{n} N(V_i|0, \sigma_V^2) \tag{3}$$

Similar to Eq.(3), through Bayesian Inference,

$$\begin{aligned} \Pr(U, V|R, \sigma_R^2, \sigma_U^2, \sigma_V^2) = & \prod_{i=1}^{m} \prod_{j=1}^{n} [N(r_{i,j}|g(U_i^T V_j), \sigma_2^R)]^{I_{i,j}^R} \\ & \times \prod_{i=1}^{m} N(U_i|0, \sigma_U^2) \times \prod_{j=1}^{n} N(I_i|0, \sigma_I^2) \end{aligned} \tag{4}$$

The learning of the model is done by posterior probability of the latent matrices U and V, that is similar to minimizing the sum-of-squares of the objective function with the regularization term for removing the overfitting.

$$\varphi(U,V) = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}(R_{ij} - U_i V_j^T)^2 + \frac{\lambda_U}{2}\left\|U\right\|_F^2 + \frac{\lambda_V}{2}\left\|V\right\|_F^2 \tag{5}$$

where $\lambda_U = \frac{\sigma_R^2}{\sigma_U^2}, \lambda_V = \frac{\sigma_R^2}{\sigma_V^2}, \lambda_Z = \frac{\sigma_R^2}{\sigma_Z^2}, \lambda_{Im} = \frac{\sigma_R^2}{\sigma_{Im}^2}$, and $||.||_F^2$ denotes the Frobenius norm. Local minima of the objective function in Eq.(4) can be obtained from the Gradient Descent Algorithm. Fig. 1 shows the conceptual representation of the basic MF approach. As shown in Fig. 1, initially, user-item preference matrix bifurcated into training and testing data, which consists of rating preference of individual users. Training data, is then fed into the MF model, which is responsible for setting of model parameters as well as cost minimization functions such as Stochastic Gradient Descent (SGD) [14]. Note that, MF model constitutes of model parameters, i.e. learning rate, number of epochs (iterations) and regularization parameter and cost minimization function. For minimization of the cost function, different methods are being used For ex: SGD, Alternative Least Square (ALS) and Particle Swarm Optimization (PSO) etc. These methods adhere to the idea of achieving local minima instead of global minima of the cost function. Resultant lower dimensional user, item latent factor matrix has been used for predicting recommendation for the users existing in the testing data.

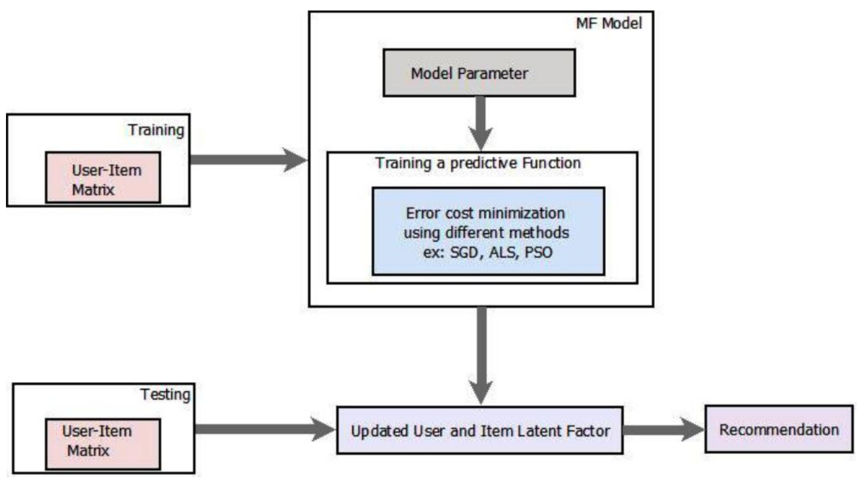


Fig. 1: A conceptual view of the Matrix Factorization

3.2. Joint Matrix Factorization for Recommendation System

As earlier in the introduction, we have discussed, Yue Shi et al. proposed a Joint Matrix Factorization (JMF) model (as mentioned in Fig. 2, for incorporating other contextual information related to the user and item instead of depending only upon the user item rate information for the improvement in accuracy. The formula for obtaining the predicted rating matrix by leveraging the user and item latent factor with the contextual (side) information about them is explained as:

$$\varphi(U,V) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} I_{ij}(R_{ij} - U_i V_j^T)^2 + \frac{\alpha}{2}\sum_{j=1}^{N}\sum_{n=1}^{N} I_{jn}^S(S_{jn} - V_j^T V_n) + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 \tag{6}$$

Here, first component shows the model dependency on user-item preference matrix, while the second component confirms it dependency also on the side information such as user's mood or movie keywords. Moreover, third and fourth components refer to regularization terms of users and items respectively. Note that, Yue Shi et al. proved its worthiness to incorporate contextual information rather than depending only on the rating information for regenerating the original matrix R. The model was tested over varying values of α. It results in higher accuracy at α = 0.0001 to 1, which justified the importance of the second component of Eq.(6). Fig. 2 represents a similar model as in previous Fig. 1, except both the user-item preference as well as contextual information is used to train the MF model. Note that, here the updated user item Latent factor matrix is implicitly influenced by contextual information rather than only depending on the user-item preference matrix.

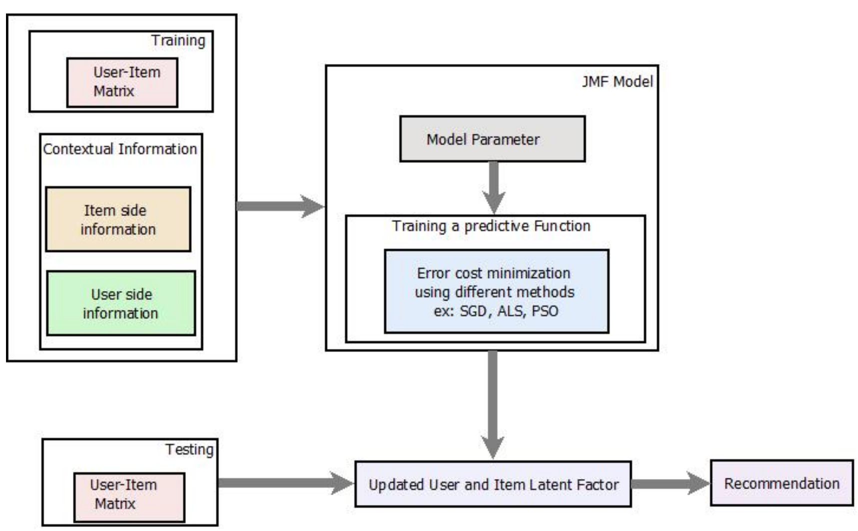


Fig. 2: A conceptual view of the Joint Matrix Factorization

3.3. Methodology used for Semantic Feature Generation

This section briefly explains about the semantic feature generation from a LOD dataset known as DBpedia. DBpedia is a RDF version of Wikipedia, which leverages "infobox" information on Wikipedia. Due to it originated from encyclopedic data, it currently has 4.58 million entities which comprise different domain information like persons, places, creative works, species and diseases.

To exploit DBpedia for generating features related to each music composer, we have stored a mirror copy of DBpedia in our local system [1]. For storing this RDF data we have used virtuoso RDF storage that allows us to interact DBpedia via SPARQL query language. Before extracting features for each music composer, we required a mapping of their names with DBpedia URIs as a seed point to start with. For that, we have used already mapped data [2] provided by Vito et al. in 2013. Using 8,177 mapped URIs of music composers we have fetched 12,000 distinct categorical features for ex: Category:Activism by issues, Category:American people of Greek descent, etc. Here, the "categorical features" specify the musical categories of a music composer. Here, "categorical features" specify the musical categories of a music composer. The information of music composer and their associated musical categories have stored in the matrix format, where rows denote music composer's URIs and columns denote musical categories. A cell of this matrix denotes the presence or absence of a particular musical category for a specific music composer with '1' and '0' respectively. A cell of this matrix denotes the importance of a particular musical category for a music composer. In this paper, we refer this matrix as item-property matrix, where Item and property represent music composers and their categories, respectively. The importance of a particular musical category has been calculated by below formula, which results in the weighted factor between '0' and '1'. This formula was initially proposed by Petar et al. [19] in 2014 for converting RDF data into a weighted matrix format to make it suitable for mining.

$$Item\,(P) = \frac{1}{f_{Item,P}} \times log\frac{N}{f_{Item,P,O}} \tag{7}$$

Here, "P" denotes the predicate of the particular item, and $f_{Item,P}$ represents the frequency of the occurrence of the predicate with the particular item. 'N' is the total number of items in the RDF graph and $f_{Item,P,O}$ shows a number of other resources having the same predicate. Note that here, item is denoted as a subject or a resource. Fig. 3 shows one sample excerpt of the RDF graph having a subject and objects as nodes and predicates are denoted as links. The weights associated with the links represent the importance of the object for the subject related to each other with a specific relation (i.e. Predicate). $F_1, F_2, ...F_n$ denote features of the seed URI "http://dbpedia.org/resource/Hopesfall". **For ex:** Space-rock is the genre of the Hopesfall band, thus act as a feature and represented by F_3 node in the Fig. 3. RDF triplet (Subject ,Predicate ,Object) representation of it can be done as follows:

```
<http://dbpedia.org/resource/Hopesfall> dbo:genre Space_rock.
```

It illustrates that Space-rock is the genre of the Hopesfall band.

[1] https://joernhees.de/blog/2014/04/23/setting-up-a-local-dbpedia-3-9-mirror-with-virtuoso-7/
[2] http://sisinflab.poliba.it/semanticweb/lod/recsys/datasets/

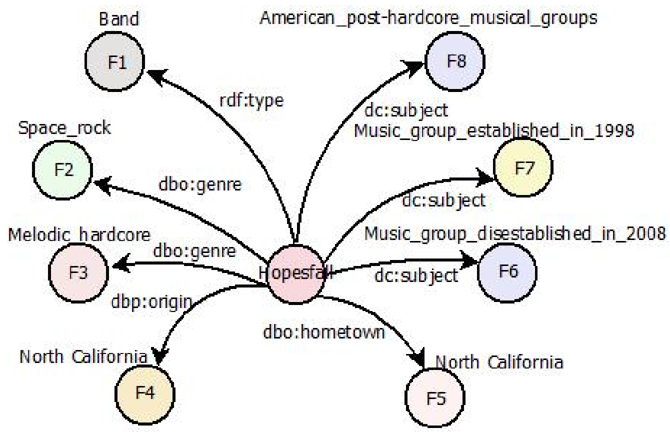


Fig. 3: An excerpt of the RDF graph for the music band "Hopesfall"

3.4. Time aware Recommendation

Recommendation based on time has been proved to improve the accuracy of the RS [14]. Heuristic and Model Based are two methodologies to utilize time variant information for producing ratings. In this paper we choose to exploit model based approaches in our effort to overcome the problems of high dimensionality and sparsity. We therefore focused only on model based time-aware system for SEM-JMF. The general formulation of the time dependent contextual RS is given as:

$$X : U \times I \times T \rightarrow R$$

where, T is the time and it can be represented in different ways like: year, month, day, hours etc.

Adding time variant contextual information, gained popularity since 2005, when Ding et al. [10] considered the exponential decay of time while building time-aware based RS. However, Korean et al. [14] proposed time aware system for rate prediction of the items excluding the effect of decay factor. Existing Time aware RS, partitioned the time in a different context like Baltrunas et al.[2] divided the data based on different variants of time, i.e. (morning, evening), (weekend, workday) and more precisely in even and odd hours. Baltrunas et al. proved that the highest improvement was obtained by considering more precise information of even and odd hours. Recently, model based time dependent RS approaches exploited MF and tensor factorization machines by Korean el al. and Xiong el al. [26]. The main disadvantage of these models is that, they are computationally expensive in nature. Nevertheless, the authors argued that these models are capable of predicting the future behavior of the users better than other existing methods. Some of the open issues in these types of RSs are, to measure the improvement of incorporating time with different evaluation conditions i.e. for specific rating distribution [5]. In this paper we explore this issue by evaluating the system on following criteria:

- Comparing of time dependent and independent system
- Evaluating the performance system on varying samples of Training and Testing data

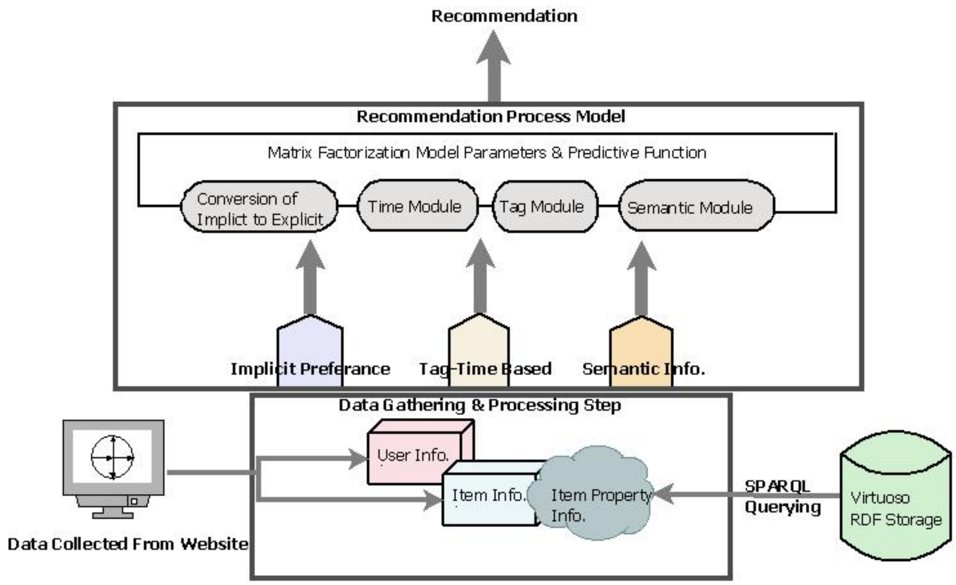


Fig. 4: Description of SEM-JMF Methodlogy

4. Proposed Work

We now turn to investigate the effect of utilizing JMF approach with the implicit preference of the users. It would be very interesting to observe the results of applying modified JMF with the Last.fm dataset. Recently, few attempts have been made to handle the implicit likings of the user. We assume that the dependent and relative contextual information will produce significant improvement in the accuracy of the proposed RSs. In this section the authors introduce a novel method for converting implicit preference of the user to explicit ratings. For that, we proposed a model (as shown in Fig. 4) to combine relative contextual information with the user-rating matrix and analyze it in different ways. As shown in Fig. 4, after collection and preprocessing of data (Last.Fm and DBpedia), the next step is to work on three main functionalities of the proposed method, namely “Implicit Preference”, “Tag-Time Based” and “Semantic information”. More specifically, these functionalities will be fulfilled from four different modules, i.e. “Conversion of Implicit to Explicit”, “Time module”, “Tag module” and “Semantic module” that later on feeding into the modified JMF algorithm for generating recommendations. These four modules are explained sequentially in this Section.

In Last.fm dataset ratings are not present explicitly, to convert this information into a fixed rating scale, we followed the method mentioned below. As given in the Last.fm dataset, the frequency for a given user *i* and Musical Artist *j* can be normalized as follows:

$$freq_{i,j} = \frac{count(i,j)}{\sum_{j'} count(i,j')} \tag{8}$$

The formula calculates play frequency for a given user i and a Musical artist j, by user’s play count for that artist normalized by his total plays. Let, $freq_k(i)$ notation refers to the frequency count of the k-th most listened artist for a user i. The rating for an artist with rank k has been computed using the linear function of the frequency percentile as given below:

$$r_{i,j} = 4 \cdot \left(1 - \sum_{k'=1}^{k-1} freq_{k'}(i)\right) \tag{9}$$

The formula will assign the rating to each Musical Artist for a user in the range of 0-4. Top 25 percentile artist will be assigned a rating in the range of 3-4, next 25 percentile will assigned as 2-3 and so on. But this approach quickly converged and induced global error in the ratings. To overcome from the problem of skewness of ratings in the dataset binning method has been applied, in which all the listened artist for a user i is assigned in non-overlapping bins on the basis of the rating they obtained from the Eq. as given below :

$$\begin{aligned} B_i^1 &= \{j : 4 \geq r_{i,j} > 3\} \\ B_i^2 &= \{j : 3 \geq r_{i,j} > 2\} \\ B_i^3 &= \{j : 2 \geq r_{i,j} > 1\} \\ B_i^4 &= \{j : 1 \geq r_{i,j} > 0\} \end{aligned} \tag{10}$$

Aggregated Error of each bin is determined by the following formula, it provides the normalized error of the overall bins:

$$e_{i,j}^{norm} = \frac{r_{i,j} - u_i^T v_j}{\sqrt{\|B_i^t\|}} \tag{11}$$

For estimating the user and the product matrices with the use of this error normalization can be written as the following, where the updation is done by gradient descent method to obtain a local minima:

$$\begin{aligned} u_i &\leftarrow u_i + \gamma \cdot \left(e_{i,j}^{norm} \cdot v_j - \lambda \cdot u_i\right) \\ v_j &\leftarrow v_j + \gamma \cdot \left(e_{i,j}^{norm} \cdot u_i - \lambda \cdot v_j\right) \end{aligned} \tag{12}$$

Here, p_j denotes $Artist \times LatentFactor$ and u_i shows $User \times LatentFactor$. Parameter γ, denotes the learning rate of the model which should be small such as 0.001. λ parameter is used for regularization so that model does not overfit over the training dataset and it will be between 0.1 to 0.2. The updation continues till convergence or maximum number of iterations of the program have achieved.

4.1. Combine Semantic Information

DBpedia is the source from where we have fetched data regarding each Musical Artist using SPARQL querying. In our case $R = User \times Artist$ and $S = Artist \times Categories$, are the Rating (R) and Semantic Information (S) matrix with the dimensions of $[1892 \times 17,632]$ & $[17,632 \times 15,000]$ respectively. Factorize matrices $V = LatentFactor \times Artist$ and $W = LatentFactor \times Category$ are initially a random matrices, taken for the MF process. As discussed in the previous sections, our goal is to exploit the information of semantic features, thus we propose following cost function:

$$\varphi(V) = \frac{1}{2} \sum_{j=1}^{N} \sum_{n=1}^{K} I_{jn}^S (S_{jn} - V_j^T W_n)^2 \tag{13}$$

Taking the influence of semantic features into account for the loss function in the basic MF model, a Semantic JMF can be formulated as:

$$\varphi(U,V) = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}^{R}(e_{i,j}^{norm})^2 + \frac{\alpha}{2}\sum_{j=1}^{N}\sum_{n=1}^{K} I_{jn}^{S}(S_{jn} - V_j^T W_n)^2 + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2) \tag{14}$$

Minimization of the objective function in Eq.(14), can be solved as the gradient descent method, which alternatively fixes the U and V matrices. The process results in a local minimum solution. The gradients of $\varphi(U,V)$ with respect to U and V can be computed as:

$$\frac{\partial\varphi}{\partial U_i} = \sum_{j=1}^{N} I_{ij}^{R}(e_{i,j}^{norm})V_j + \lambda U_i \tag{15}$$

$$\frac{\partial\varphi}{\partial V_j} = \sum_{i=1}^{M} I_{ij}^{R}(e_{i,j}^{norm})U_i + 2\alpha\sum_{n=1}^{N} I_{jn}^{S}(V_j^T W_n - S_{jn})W_n + \lambda V_j \tag{16}$$

So, for updation of the user and item property vector of Eq.(12), we obtained the equations as below:

$$\begin{aligned} u_i &\leftarrow u_i + \gamma \cdot \left(e_{i,j}^{norm} \cdot v_j - \lambda \cdot u_i\right) \\ v_j &\leftarrow v_j + \gamma \cdot \left(e_{i,j}^{norm} \cdot u_i + 2\alpha\left(S - v_j w_n\right) - \lambda \cdot v_j\right) \end{aligned} \tag{17}$$

where v_j represents Category $\times LatentFactor$, v_j, w_n denotes $LatentFactor \times Artist$, and denotes S denotes $Artist \times Category$. Here, α term decides the importance of the combination of Semantic information.

4.2. Combine Artist Tag Information

As aforementioned, other contextual information that we are interested to incorporate is the tag information, given by the user to the Musical artist in the different periods of time. In this section, the authors explore retrieval and utilization of the tags (for more detail see [3]). Total one thousand tags have taken and others are discarded due to the repetitions and incoherencies in it. To use this information we have proposed the following equation:

$$\begin{aligned} u_i &\leftarrow u_i + \gamma \cdot \left(e_{i,j}^{norm} \cdot v_j - \lambda \cdot u_i\right) \\ v_j &\leftarrow v_j + \gamma \cdot \left(e_{i,j}^{norm} \cdot u_i + 2\alpha\left(S - v_j w_n\right) + 2\beta\left(T - v_j g_k\right) - \lambda \cdot v_j\right) \end{aligned} \tag{18}$$

Note that, in comparison to Eq.(17), Eq.(18) adds one more term, i.e. β, which influences the reconstructed user-item preference matrix with the artist's tag information. Here, g_k denotes $Tag \times LatentFactor$ and v_j denotes $Artist \times LatentFactor$. Matrix T, shows $Artist \times Tag$ in which rows and columns represent an artist and their given tags, respectively. Moreover, $t_{j,k} \in$ T is calculated using the following formula:

$$t_{j,k} = \frac{1}{a}log(\frac{N}{b}) \tag{19}$$

[3] https://github.com/nidhikush/Sem-JMF-LastFM/

where, N= total no. of tags presents in the dataset, a= no. of times users has given these tags (how often this tag has been used by other Artist) and b= no. of users who have been given the same tag to the same Artist.

More specifically, processing T enables us to exploit the user's tag selection in reconstructing user-item preference matrix. It fulfills our assumption of incorporating artist tag information in this section.

4.3. Adding Time Information for Tagged Artist

Addition of time information is based upon the assumption of the decaying importance of user assigned tags with respect to time. Data acquisition and processing of time are explained with code [4].

To combine tag information we refer to Tereza et al. approach [12], that utilizes the time decaying factor for predicting recommendations. The mathematical formulation of it is given as follows:

$$postScore_i = \lambda^{\triangle Time_i} \tag{20}$$

where λ is the time decaying function which is smaller than '1', we have taken λ=0.9.
To provide weightage to each tag i we have to calculate tag specificity, by the given formula:

$$TagSpecificity = log(50 + tagCount_i) \tag{21}$$

and to measure the overall tag score for individual tag i, we have used following formula:

$$TagScore_i = \frac{\sum_i^n (PostScore)}{TagSpecificity_i} \tag{22}$$

Let's suppose Tag Score is denoted as $c_{i,j}$. We assume Tag Score as the confidence coefficient for the Matrix T ($Tag \times LatentFactor$), so to include this coefficient into the proposed equation we modified the Eq.(18) as follows:

$$\begin{aligned} u_i \leftarrow u_i + \gamma \cdot \left(e_{i,j}^{norm} \cdot v_j - \lambda \cdot u_i\right) \\ v_j \leftarrow v_j + \gamma \cdot \left(e_{i,j}^{norm} \cdot u_i + 2\alpha \left(S - v_j w_n\right) + 2\beta \times c_{i,j} \left(T - v_j g_k\right) - \lambda \cdot p_j\right) \end{aligned} \tag{23}$$

Note that, in Eq. (23) we multiplied the obtained Tag Score for tag i to the β factor for boosting the influence of reconstructing user-item preference matrix with decaying time factor.

5. Implementation Detail and Analysis

5.1. Dataset Description

We have evaluated SEM-JMF method on Last.fm dataset [5] and DBpedia [6]. Last.fm dataset [6] contains 1,892 users, 17,632 artists, 11,946 tags. The dataset also has the information about timestamps, which was gathered when the tag assignments were done by each

[4] https://github.com/nidhikush/Sem-JMF-LastFM/
[5] http://grouplens.org/datasets/hetrec-2011/
[6] http://datahub.io/dataset/dbpedia

Algorithm 1 Basic SEM-JMF Approach

Input : User-Artist implicit rating matrix **M**, Artist-Category specific matrix **S**, Tag-Artist matrix **T** and Time-Tag matrix **C**, tradeoff parameter α, β, regularization parameter λ, stopping condition η, ϵ

Output: predicted matrix $\hat{R}$

Initialize $U^{(0)}$, $V^{(0)}$ with random values and t=0, f=0, η=maximum number of iterations, ϵ=minimum error difference.

Convert implicit rating into explicit rating using Eq.(8) & (9).

Calculate $\varphi^{(t)}$ as in Eq.(14).

repeat

  $i = 1$

  Calculate $\frac{\partial\varphi}{\partial U^{(t)}}$, $\frac{\partial\varphi}{\partial V^{(t)}}$ as given in Eq.(15) and (16)

  repeat

    $i = \eta$ // stopping condition-1 (indicator of number of iterations)

  until $\varphi(U^{(t)} - \eta\frac{\partial\varphi}{\partial U^{(t)}}, V^{(t)} - \eta\frac{\partial\varphi}{\partial V^{(t)}}) < \varphi^{t}$

  $U^{(t+1)} = U^{(t)} - \eta\frac{\partial\varphi}{\partial U^{(t)}}$, $V^{(t+1)} = V^{(t)} - \eta\frac{\partial\varphi}{\partial V^{(t)}}$

  Calculate $\varphi^{(t+1)}$ as in the Eq.(14)

  if $1 - \varphi^{(t+1)}/\varphi^{(t)} \leq \epsilon$ **then**

    $f = 1$ // stopping condition-2 (indicator of convergence)

  end

  $t = t + 1$

until $f = 1$

$\hat{R}$=$U^{(t)T}V^{(t)}$

Table 1: Dataset-I Statistics

(a) Last.Fm

Features	Values
No. of users	1892
No. of items	17,632
Rating matrix Density	2.783×10^{-3}
No. of tag statements	186,479

(b) Features obtained by mapping of DBpedia & Last.Fm (8146 Artist)

DBpedia Property	Values	Sparsity (in %)
dcterm:Categories	12,101	0.9837854

Table 2: Dataset-II Statistics

(a) MovieLens

Features	Values
No. of users	71,567
No. of items	10,681
Rating matrix Density	4.468 %

(b) Features obtained by mapping of DBpedia & MovieLens1M (3091 Movies)

DBpedia Property	Values	Sparsity (in %)
dcterm:Categories	10,940	0.998

particular user. It was released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Hetrec 2011. In Last.fm dataset (as shown in Table. 1a), there is no explicit ratings present. The ratings are available in the form of user listening count for the artist, which ranges from 1 to 352,698. To test our method we map the listening count into ratings by the proposed method and introduce a novel method of associating the semantic features for recommendations as described in Section 4. For semantic features, DBpedia dataset provides RDF information of items which needs to be converted into a processable format Section 3.3 details about the conversion of RDF graph into weighted item-property matrix which denotes the importance of the properties for a particular artist. For item-property matrix we have used categorical information of each musician, the file have more than 12,000 attributes with 0.9837 % sparsity. Table. 1b shows the feature values obtained from the mapping of DBpedia-Last.fm, used in this work. Here, "dcterm:Categories" is nothing but, a well known standard vocabulary for defining the categorical information of the entities in RDF datasets. More Specifically, in our case it defines the categories of Musical Artist of DBpedia dataset.

Moreover, although the proposed approach had been developed for implicit user preference, for proving its applicability also with explicit user preference, we have chosen another dataset i.e. MovieLens dataset [11]. This dataset was collected by GroupLens research groups of the University of Minnesota. It contains one million ratings provided by 71,567 users for 10,681 movies (see Table. 2a). Ratings are provided in the predefined scale of 0-5, in which "1" shows the lowest, "5" represents the highest rating. However, "0" denotes unwatched or missing information by users. Analogous to Last.Fm and DBpedia, mapping of MovieLens and DBpedia[7] also has been used for generation of movie categories. Using SPARQL queries we have fetched 10,940 distinct movie categories as shown in Table. 2b. These movies and their categorical information have been converted into a weighted item-property matrix and have stored in the similar format as discussed in Section 3.3.

5.2. Experimental Analysis

The basic procedure followed for SEM-JMF is explained in algorithm 1. The code is made available the under GPLv3 license [8] in python. To prove the correctness of SEM-JMF, we have evaluated the results using Root Mean Square Error (RMSE) measure, which defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{i=N}(R_{i,j} - u_i^T v_j)^2}{N}} \tag{24}$$

here, $R_{i,j}$ is the original rating given by user i to the item j, while $u_i^T v_j$ is the predicted rating from the proposed model and 'N' is the total number of testing set. It basically calculates the difference of original and predicted ratings of user-item preference matrix. Note that, here the predicted ratings have been calculated using the SEM-JMF method proposed in this paper. Lesser the value of RMSE represents better the performance of the method.

SEM-JMF is tested over many iterations to find the suitable value of λ, γ for which the algorithm showed the least RMSE value. As shown in Fig. 5, in implicit preference setup, the average user only listened to a small fraction of artists, thus makes it highly skewed. Normal rate assigning will not work here to convert implicit into explicit preference due to

[7] http://sisinflab.poliba.it/semanticweb/lod/recsys/datasets/
[8] https://github.com/nidhikush/Sem-JMF-LastFM/

skewness in data. For alleviating this problem normalization of user preference is needed. The first module of SEM-JMF converts the implicit preference of users to the ratings with the help of binning method as described in Section 4. The binning process converts listening count into fixed rating scale (normalize) between 0-4, thus for each user there are four bins that covers all the rated artist by him/her. Our goal is to predict ratings to an unrated artists for a target user. In our case, after normalization the sparsity of the matrix is 0.981037%.

Fig. 6, shows the RMSE obtained by applying Stochastic gradient descent (SGD) in both the cases with constant parameters. SGD with normalization has a less cumulative RMSE and performs better in almost all the cases, i.e. 0-25%,25-50% and 50-75%, thus it shows the importance of the normalization method discussed in Section 4. Average number of frequencies in four bins in different training splits (i.e. 80%, 60%, 40%, 20%) has been shown by Fig. 7a. It shows that most of the artists assigned ratings in the range of 3 to 4.

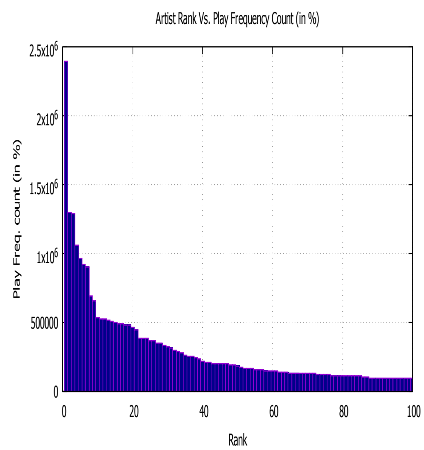


Fig. 5: First top 100 ranked artists are shown with their play frequency.

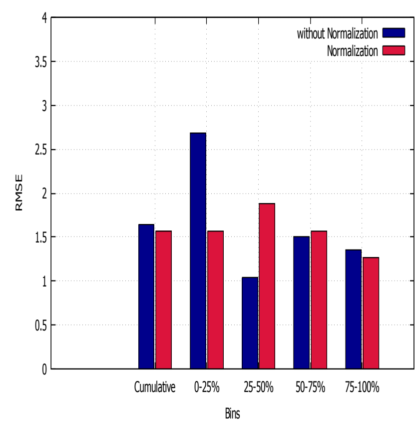
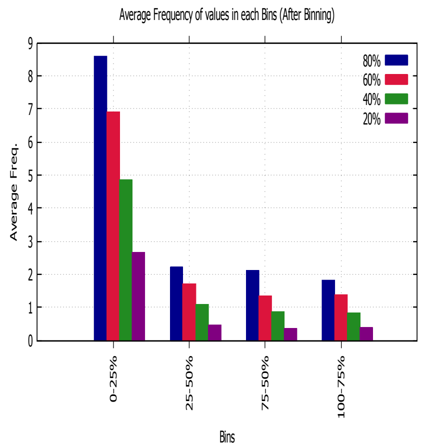
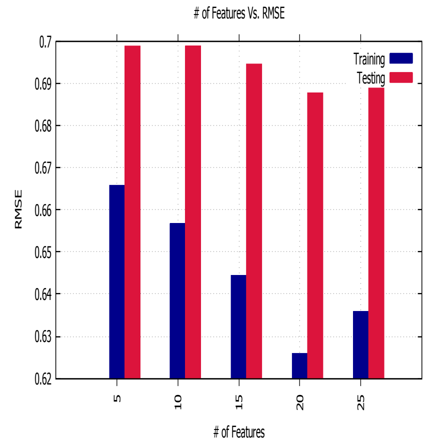


Fig. 6: Percentile RMSE achieved by with SGD and without SGD. Parameters for both are γ=0.003, λ=0.001 and Latent Factor (d)= 5

After binning, SGD has been tested with different features with constant parameters, to see the variance of RMSE values on different Latent Factors with Fig. 7b. Variance in RMSE values shows the dependency of features on it. Variance of learning rate γ and regularization λ with respect to RMSE have also been analyzed and shown by Fig. 10. Convergence of SGD with training and testing data with different epoch is shown by Fig. 11. The parameters used for second, third and fourth modules shows the importance of each module in reducing the cost function in SGD algorithm. Fig. 12 shows the phenomena of JMF, where each parameter has some effect on reducing the error of the overall system. It is induced from the graph that, α (i.e. Semantic) and β (i.e. Semantic+Tag) parameters have greater significance in reducing the RMSE values than the parameter $C_{(i,j)}$



(a) After binning, average number of frequencies in each of the four (i.e. 80%, 60%,40%,20%) Training splits. In all Training splits most of the artists assigned a rating in the range 3-4.



(b) Training and testing errors of the SGD for different values of Latent Factors (d). The performance is improved as a number of features are increased till a point.

Fig. 7: Analysis after binning preprocess

of the fourth module (i.e. Semantic+Tag+Time). Training and Testing splits are randomly selected 100 times and passed in different modules to identify which module can work superior among them. Fig. 13 clearly shows improvement in model in case of semantic and tag inclusion of the data with normalized user preference over semantic and Tag time.

In order to prove the applicability of SEM-JMF algorithm, it also has been evaluated with an explicit dataset i.e. MovieLens1M. Implementation code [9] is made openly avialable for research purpose. The details of a dataset are mentioned in Section 5.1. For the evaluation we have chosen user-item preference matrix of MovieLens data which has rating information for a particular user-item pair. Furthermore, contextual (side) information about DBpedia categories related to each mapped item (movie) has been exploited as a item-property matrix. As discussed earlier, similar to Eq.(14) both user-item preference and item-property matrices for MovieLens dataset was processed using SGD. We have chosen Mean Absolute Precision (MAP) metric [7] for the evaluation in this case. This formulation was also used by [21] for evaluating the performance of the JMF model. The main intuition for this analysis is to justify the SEM-JMF performance with MAP. MAP was defined as follows:

$$MAP = \frac{1}{K}\sum_{u=1}^{K}\frac{\sum_{j=1}^{Nu} relv(j) \times Pu@j}{\sum_{j=1}^{Nu} relv(j)} \tag{25}$$

where, K represents the number of users in testing data and Nu is the number of recommended movies for a user u. A binary indicator $relv(j)$ is used for assigning, '1' when

[9] https://github.com/nidhikush/MovieLens1M_Sem-JMF/

rank j is relevant to user u, and equal to '0' otherwise. Precision at top j recommended items for a user u has been represented by $Pu@(j)$. MAP shows the quality of the SEM-JMF method by averaging the top-N precision of recommended items for each user in the testset. The large number of relevant items in the top-N recommendation list results in a higher value of MAP and thus ensures better performance of the system. Greater the value of MAP shows better the perfomance of the model. For processing and generating recommendation from MovieLens 1M dataset we have used a similar formulation as represented in Eq.(14). Note that, in Eq.(14) second component states the importance of incorporating semantics in maximizing the MAP of the model. For convenience, we have denoted α factor of Eq.(14) with θ symbol.To see its effect on MAP, θ symbol has been evaluated over different values between 0-50. Fig. 8 & Fig. 9 obtained from the model generated with Latent Factor =10 and # of iterations=50. It shows the model achieves highest MAP when θ parameter is between 0.0001 and 0.001. It concludes the importance of semantic parameter by the increment of 0.01 (approx.) in MAP values.

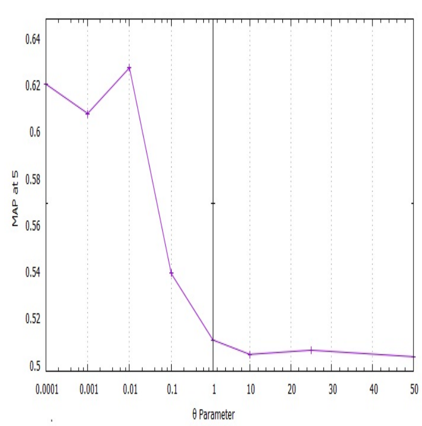


Fig. 8: MAP at 5 Vs. θ parameter.

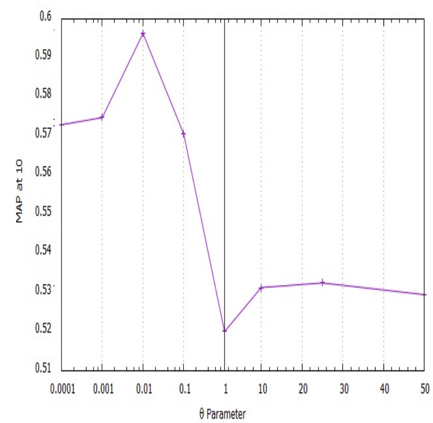
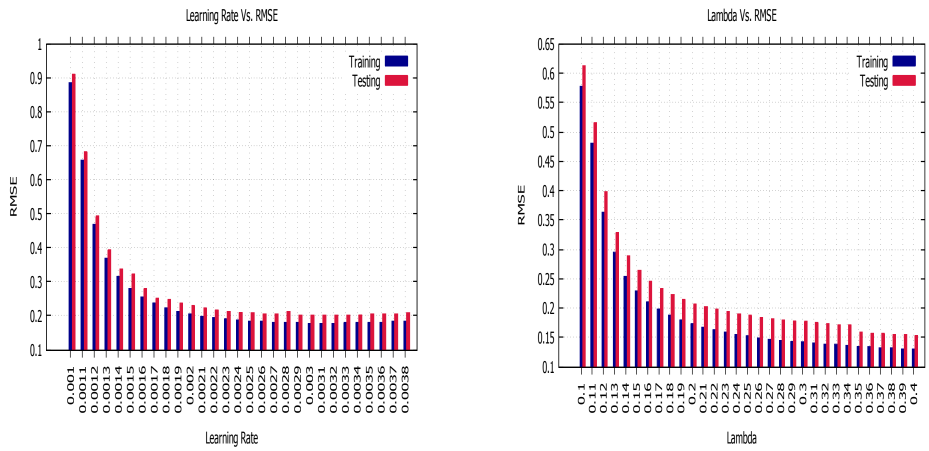


Fig. 9: MAP at 10 Vs. θ parameter.

This section explores the analysis of SEM-JMF method on two different datasets, both the analysis concludes the importance of semantic inclusion for better rate prediction. The next section will demonstrate further analysis and comparisons of the proposed approach with non-semantic methods. We have compared Last.Fm dataset with recent past algorithms[16][20] and MovieLens1M has been compared on the basis of semantic and non-semantic recommendation.

5.3. Comparison with other methods

The effectiveness of the SEM-JMF method has been verified by comparing it with two recent algorithms [16][20]. We noticed that our method significantly differs from the methods used for preprocessing and proposed for LOD based RS. Thus we choose to compare SEM-JMF with the methods proposed for incorporating social information given in



(a) Training and testing errors of the normalized SGD for different learning rates γ. The limited overfitting behavior can be controlled by varying γ.

(b) Training and testing errors of the normalized SGD for different regularization parameter λ. The limited overfitting behavior can be controlled by varying λ.

Fig. 10: Behaviour of Regularization and Learning parameter Vs. varying RMSE

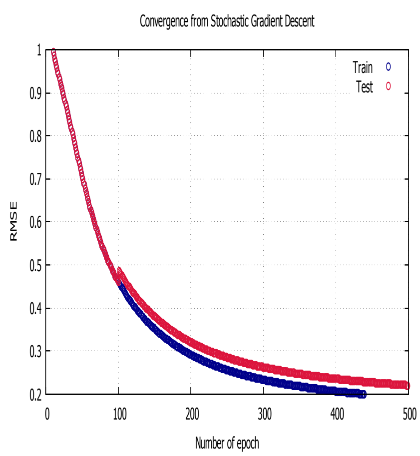


Fig. 11: Convergence profile of a SGD with explicit feedback (after binning) on an Implicit Last.Fm dataset. Constant parameters are γ= 0.003, λ= 0.001, number of Epoch= 300 and Latent Factor (d)= 5. Final errors are 0.2369490 training and 0.2569403 for testing.

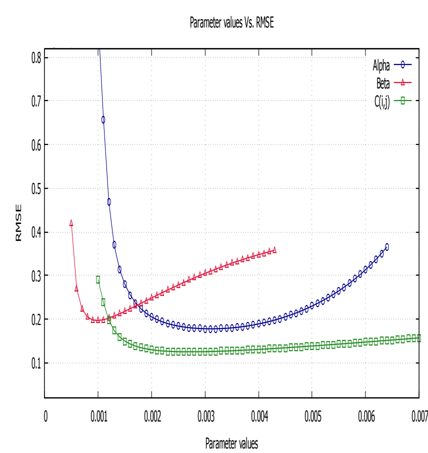
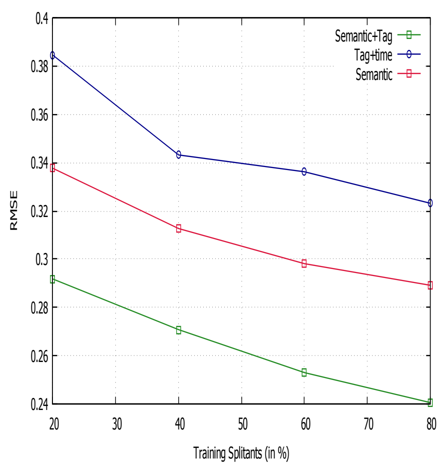
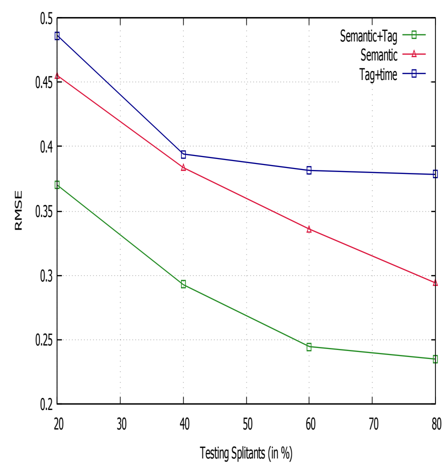


Fig. 12: Behaviour of importance factor alpha, beta and gamma for Semantic, Semantic+Tag and Semantic+Tag+time respectively Vs. RMSE. Constant parameters are γ= 0.003, λ=0.001, number of Epoch= 300 and Latent Factors (d)=5.

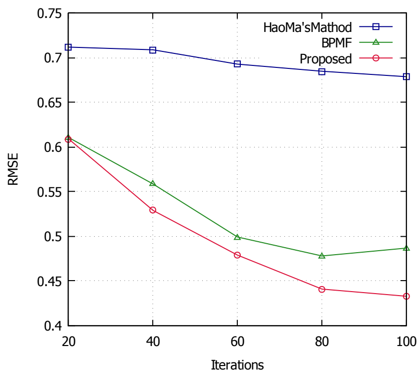


(a) Training splits of different module (Semantic, Semantic+Tag and Tag+Time) Vs. RMSE has been shown. Constant parameters are γ= 0.003, λ=0.001, number of Epoch= 300 and Latent Factors (d)=5. Semantic+Tag module behaves superior among them.

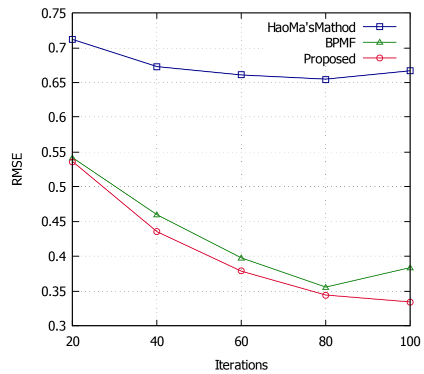


(b) Testing splits of different module (Semantic, Semantic+Tag and Tag+Time) Vs. RMSE has been shown. Constant parameters are γ= 0.003, λ=0.001, number of Epoch= 300 and Latent Factors (d)=5. Semantic+Tag module behaves superior among them.

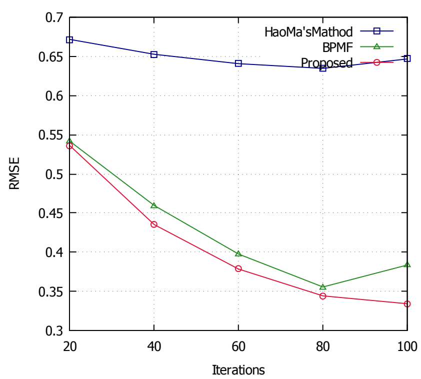
Fig. 13: Different modules Vs. RMSE



(a) Testing RMSE for 40% of all methods with all modules on Last.fm dataset, the x-axis shows the number of epoch and y-axis shows testing RMSE.

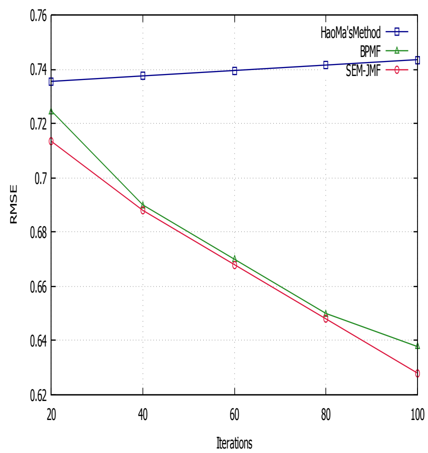


(b) Testing RMSE for 60% of all methods with all modules on Last.fm dataset, the x-axis shows the number of epoch and y-axis shows testing RMSE.

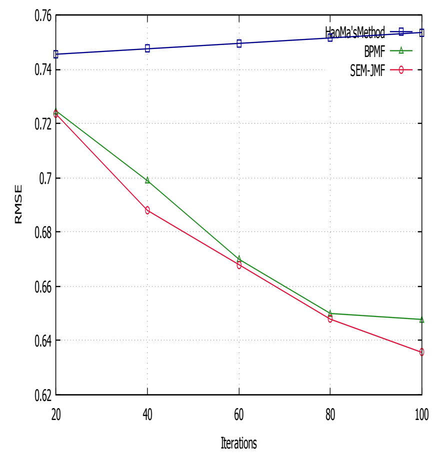


(c) Testing RMSE for 80% of all methods with all modules on Last.fm dataset, the x-axis shows the number of epoch and y-axis shows testing RMSE.

Fig. 14: Comparison of Proposed method with other methods (Last.Fm dataset)



(a) Testing RMSE for 10% of Cold start Users with the constraints of γ= 0.003, λ=0.001 and Latent Factors (d)=5.



(b) Testing RMSE for 10% of Cold start Items with the constraints of γ= 0.003, λ=0.001 and Latent Factors (d)=5.

Fig. 15: Comparison of Proposed method with other methods in Cold Start User/Item

[16][20]. Note, that in this section we considered all the four modules as our proposed method and compare its results with other algorithms.

The result of the comparisons represented from the Table. **??**, shows SEM-JMF performs better in some conditions with the constraints of λ= 0.001, γ= 0.003, Latent Factor (d)= 5 and number of epoch= 300. Evaluation has been done on different splits of training & testing data, see in Fig. 14.

For testing the SEM-JMF algorithm in cold start user and item settings, we randomly select 10% users and 10% items and treated them as testing data. All the ratings of these cold start user/item are treated as testing data and others are used for training data. Fig. 15 shows SEM-JMF performs slightly better than BPMF.

Table. 4 shows the performance comparison of SEM-JMF on MovieLens1M dataset. As discussed earlier that the θ parameter defines the importance of the semantic inclusion in the basic MF method. Note that, at θ= '0', the algorithm behaves as a simple MF approach. We observed with different training set splits and concludes that SEM-JMF performs best at θ=0.01, as shown in the Table. 4.

6. Conclusion & Future Outlook

In this paper, based on the intuition that the side information of the items and time variant information of users will affect the behavior of the recommendation, we introduced a new mathematical model for combining different side information with explicit user preferences for predicting recommendation. While the inclusion of various side information, the authors also analyze the importance of combining them individually. For side information of items the proposed work utilizes semantic features, extracted from DBpedia. To ob-

Table 3: Comparisons for Last.Fm dataset

Dimensions	**Splits**	**Hao Ma' Method [16]**	**BPMF[20]**	**SEM-JMF**
10	40%	0.6936	0.4665	0.4698
	60%	0.6529	0.4502	**0.4476**
	80%	0.6210	0.4465	**0.4443**
30	40%	0.6977	0.4686	0.4845
	60%	0.6708	0.4529	0.4678
	80%	0.6418	0.4467	**0.4390**

Table 4: Comparisons for MovieLens1M dataset

Size	**Parameters (number of Epoch=50, Latent Factor=10)**	**JMF (Non-Semantic)**	**SEM-JMF(with Semantics) θ=0.01**
80%	MAP@5	0.6336	**0.6435**
	MAP@10	0.5707	**0.5906**
	Precision@5	0.9054	**0.9576**
	Precision@10	0.9468	**0.9889**
60%	MAP@5	0.6256	**0.6432**
	MAP@10	0.5679	**0.5809**
	Precision@5	0.8938	**0.9321**
	Precision@10	0.9183	**0.9498**

serve the temporal effects our model used the timestamps when the tag assignments were done. This information is fetched from Last.fm dataset as described earlier in Section 5.1. Our work can be seen as the first step towards for combining semantic side information and time variant information with the implicitly defined user-item preference information using MF approach. The method is advantageous as it helps to utilize semantic features and overcomes the problem of preprocessing tasks for extracting features from textual information. Analysis of SEM-JMF on Last.fm shows Semantic and Semantic+Tag added significant benefits for accurate prediction rather than Semantic+Tag+Time. We have empirically tested the algorithm proposed in this work with two different datasets, as stated in the Section 5.1 & 5.2. Furthermore, to prove the appropriateness of the model we have also analyzed the method with MovieLens1M dataset which explicitly defines the user-item preference. The observation shows its worthiness with the improvement of 0.01 in comparison to non-semantic based approach. When fusing the semantic information of items and time variant information of users given tag, we ignore the semantic relationship among the tags assigned by users. Consequently, our future work will be to make the model which can leverage semantic information more intelligently by considering more semantic databases and test the algorithm with different accuracy measures.

References

1. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. ACM, New York, NY, USA (2010)
2. Baltrunas, L., Amatriain, X.: Towards time-dependant recommendation based on implicit feedback. In: In Workshop on context-aware recommender systems CARS2009 (2009)

3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. 5(3), 122 (2009)
4. Campbell, L.M., MacNeill, S.: The semantic web. linked and open data. In: JISC Cetis center for educational technology and iteroperability standards. pp. 1–6. JISC CETIS (June 2010)
5. Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols. User Modeling and User-Adapted Interaction 24(1-2), 67–119 (Feb 2014), http://dx.doi.org/10.1007/s11257-012-9136-x
6. Cantador, I., Brusilovsky, P., Kuflik, T.: 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In: Proceedings of the 5th ACM conference on Recommender systems. RecSys 2011, ACM, New York, NY, USA (2011)
7. David Wood, Marsha Zaidman, L.R., Hausenblas, M.: In: Fagerberg, J., Mowery, D., Nelson, R. (eds.) Linked Data-Structured data on the Web. Manning Publications, Manning Publications Co, 20 Balwin road, Shelter Island, NY 11964 (2013)
8. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D.: Exploiting the web of data in model-based recommender systems. In: Proceedings of the Sixth ACM Conference on Recommender Systems. pp. 253–256. RecSys '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2365952.2366007
9. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D., Zanker, M.: Linked open data to support content-based recommender systems. In: Proceedings of the 8th International Conference on Semantic Systems. pp. 1–8. I-SEMANTICS '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2362499.2362501
10. Ding, Y., Li, X.: Time weight collaborative filtering. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management. pp. 485–492. CIKM '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1099554.1099689
11. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. 5(4), 19:1–19:19 (Dec 2015), http://doi.acm.org/10.1145/2827872
12. Iofciu, T., Demartini, G.: Time based tag recommendation using direct and extended users sets. In: Eisterlehner, F., Hotho, A., Jschke, R. (eds.) ECML PKDD Discovery Challenge 2009 (DC09). vol. 497, pp. 99–107. CEUR Workshop Proceedings, Bled, Slovenia (September 2009), http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-497/
13. Jannach, D., Zanker, M., Felfering, A., Friedrech, G. (eds.): Recomemnder System An Introduction, vol. 1. Cambridge University Press,Cambridge, United Kingdom (2011)
14. Korean, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (Aug 2009), http://dx.doi.org/10.1109/MC.2009.263
15. Liu, J., Wu, C., Liu, W.: Bayesian probabilistic matrix factorization with social relations and item contents for recommendation. Decision Support Systems 55(3), 838–850 (2013)
16. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: Social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management. pp. 931–940. CIKM '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1458082.1458205
17. Oord, A., Dieleman, S., Schrauwen, B.: Deep content-based music recommendation. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 26, pp. 2643–2651 (2013), http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/1239.pdf
18. Passant, A.: dbrec- music recommendations using dbpedia. In: Patel-Schneider, P., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J., Horrocks, I., Glimm, B. (eds.) The Semantic Web-ISWC 2010, Lecture Notes in Computer Science, vol. 6497, pp. 209–224. Springer Berlin Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-17749-1_14

19. Ristoski, P., Paulheim, H.: A comparison of propositionalization strategies for creating features from linked open data. In: Proceedings of the 1st Workshop on Linked Data for Knowledge Discovery co-located with ECML PKDD 2014 (sep 2014), http://ceur-ws.org/Vol-1232/paper1.pdf
20. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: Proceedings of the 25th International Conference on Machine Learning. pp. 880–887. ICML '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1390156.1390267
21. Shi, Y., L.M., Hanjalic, A.: Mining contextual movie similarity with matrix factorization for context-aware recommendation 4(1) (2013), http://doi.acm.org/10.1145/2414425.2414441
22. Shi, Y., Larson, M., Hanjalic, A.: Movie similarity with matrix factorization for context-aware recommendation. ACM Trans. Intell. Syst. Technol. 4(1), 16:1–16:19 (Feb 2013), http://doi.acm.org/10.1145/2414425.2414441
23. Shi, Y., Serdyukov, P., Hanjalic, A., Larson, M.: Nontrivial landmark recommendation using geotagged photos. ACM Trans. Intell. Syst. Technol. 4(3), 47:1–47:27 (Jul 2013), http://doi.acm.org/10.1145/2483669.2483680
24. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 650–658. KDD '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1401890.1401969
25. Varga, B., Groza, A.: Integrating dbpedia and sentiwordnet for a tourism recommender system. In: Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on. pp. 133–136 (aug 2011)
26. Xiong, L., Chen, X., Huang, T., Schneider, J.G., Carbonell, J.G.: Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA. pp. 211–222 (2010), http://dx.doi.org/10.1137/1.9781611972801.19
27. Zhen, Y., Li, W.J., Yeung, D.Y.: Tagicofi: Tag informed collaborative filtering. In: Proceedings of the Third ACM Conference on Recommender Systems. pp. 69–76. RecSys '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1639714.1639727