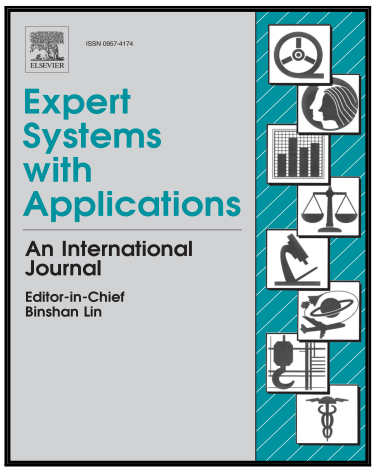


Resolving Data Sparsity and Cold Start Problem in Collaborative Filtering Recommender System Using Linked Open Data

Senthilselvan Natarajan , Subramaniyaswamy Vairavasundaram , Sivaramakrishnan Natarajan , Amir H. Gandomi

PII: S0957-4174(20)30073-7
DOI: https://doi.org/10.1016/j.eswa.2020.113248
Reference: ESWA 113248

To appear in: *Expert Systems With Applications*

Received date: 8 October 2019
Revised date: 2 January 2020
Accepted date: 24 January 2020

Please cite this article as: Senthilselvan Natarajan , Subramaniyaswamy Vairavasundaram , Sivaramakrishnan Natarajan , Amir H. Gandomi , Resolving Data Sparsity and Cold Start Problem in Collaborative Filtering Recommender System Using Linked Open Data, *Expert Systems With Applications* (2020), doi: https://doi.org/10.1016/j.eswa.2020.113248

HIGHLIGHTS

- A MF model with Linked Open Data is developed to handle Data Sparsity issue
- Hidden Data and LOD similarity measure are integrated to enhance recommendations
- The proposed framework can be applied to any domain for recommendations
- Experiments were done on Netflix and Movie Lens datasets for validation

Resolving Data Sparsity and Cold Start Problem in Collaborative Filtering Recommender System Using Linked Open Data

Senthilselvan Natarajan, School of Computing, SASTRA Deemed University, Thanjavur, India. senthilselvan@cse.sastra.ac.in

Subramaniyaswamy Vairavasundaram*, School of Computing, SASTRA Deemed University, Thanjavur, India. vsubramaniyaswamy@gmail.com

Sivaramakrishnan Natarajan, School of Computing, SASTRA Deemed University, Thanjavur, India. sivamds@gmail.com

Amir H. Gandomi, Faculty of Engineering & Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia. gandomi@uts.edu.au

*Correspondence: vsubramaniyaswamy@gmail.com

Abstract

The web contains a huge volume of data, and it's populating every moment to the point that human beings cannot deal with the vast amount of data manually or via traditional tools. Hence an advanced tool is required to filter such massive data and mine the valuable information. Recommender systems are among the most excellent tools for such a purpose in which collaborative filtering is widely used. Collaborative filtering (CF) has been extensively utilized to offer personalized recommendations in electronic business and social network websites. In that, matrix factorization is an efficient technique; however, it depends on past transactions of the users. Hence, there will be a data sparsity problem. Another issue with the collaborative filtering method is the cold start issue, which is due to the deficient information about new entities. A novel method is proposed to overcome the data sparsity and the cold start problem in CF. For cold start issue, Recommender System with Linked Open Data (RS-LOD) model is designed and for data sparsity problem, Matrix Factorization model with Linked Open Data is developed (MF-LOD). A LOD knowledge base "DBpedia" is used to find enough information about new entities for a cold start issue, and an improvement is made on the matrix factorization model to handle data sparsity. Experiments were done on Netflix and MovieLens datasets show that our proposed techniques are superior to other existing methods, which mean recommendation accuracy is improved.

Keywords: Collaborative Filtering, Matrix Factorization, Linked Open Data, Recommender System, Data Sparsity

1. Introduction

Information filtering or recommender system (RS) becomes essential and important in social network applications and electronic commerce. Websites like Flipkart, Amazon, and YouTube, have already been using filtering systems to offer personalized goods and services to their consumers. Valuable and timely recommendations can develop user fulfillment and faithfulness. Recommender systems are classified into three categories: collaborative filtering RS, content-based RS, and hybrid RS. The content-based method provides recommendations that depend on the user profile and the similarity of the item description. CF makes recommendations to users based on other user opinions. It can make complex recommendations also because, during the recommendation process, it won't bother about the content of the products. Because of this property collaborative filtering (CF) becomes a popular filtering method and plays a vital role in many applications. A hybrid system is an integration of collaborative filtering and content-based system. Since CF is a popular and widely used recommender model in many applications, it is considered here, but it has its own demerits that are addressed here. The main problem in collaborative filtering (CF) recommender method is data sparsity and the cold start issue (Najafabadi et al., 2019). Without complete information, it is hard for the CF model to recommend efficiently. A sparsity problem arises due to user interactions with a small portion of items in the particular application domain. For sample, consider MovieLens datasets which contain a rating matrix where users give ratings to movies. The rating matrix is not completely filled, only 10% of the matrix has ratings. With this available sparse data, it is difficult for CF to make a better recommendation. Another issue in collaborative filtering is the cold start issue. This is due to the lack of data about new entities, i.e., a new item/new user. Whenever a new user was added to the system, she or he had rated nothing, and the system can't compute similarity to other users. In the same case for a new product, the system can't recommend it before receiving a rating for it. Fortunately, the development of LOD helps the recommender system (RS) to acquire the required information from the LOD cloud for data-intensive tasks. Resource description framework (RDF) is the data model used in LOD. Data published on the web as RDF format and interlinking of related data was done through LOD (Oliveira et al., 2017). DBpedia is the largest knowledge base covering almost all areas of knowledge, including many entities and their relationships, and it's available free of cost, which is suitable for RS applications (Lnenicka & Komarkova, 2019; Di Noia et al., 2016). Matrix factorization (MF) is a widely used mechanism in computer applications that are incorporated herewith the collaborative filtering

model (Wang et al.,2019). It is capable of identifying the key features (dimensionality reduction) and latent factors (Nilashi et al.,2018). Nonnegative matrix factorization method seems to be an effective one, but it takes more time for the computation process (Del Corso & Romani, 2019). Cluster-based matrix factorization is adopted to solve the cold start issue in recommender systems (Hsieh et al., 2017) but takes more time. Singular Value Decomposition (SVD) is one of the matrix factorization technique widely used. After the introduction of the SVD model, many extended versions are developed to handle the data sparsity problem. They are context-aware recommendation algorithm with two-level SVD (Cui et al.,2018), co-factorization SVD model is introduced to enhance the single data source and mitigate the over-fitting problem in matrix factorization (Luo et al., 2019), incremental SVD utilizes a folding-in technique for scalability (Sarwar et al., 2000), artificial neural network with SVD performs binary predictions but not specific ratings (Billsus & Pazzani, 1998), SVD with demographic improves recommendation performance (Vozalis & Margaritis, 2007), semantic SVD++ includes user interest in the MF process (Rowe, 2014) and imputation-SVD (Yuan et al., 2019). The above said methods missed semantic data which is helpful to increase the recommendation accuracy. To overcome sparsity and cold start problems, in this research work, a new model is developed called RS-LOD for CF, which uses the LOD cloud to collect information about new entities (cold start) and also extends item-user factor vectors in the matrix factorization process (MF-LOD) to solve the data sparsity problem. The user vector is elongated by hidden feedback data and each item is elongated by semantic similar items. Semantic features from LOD are used in the MF model to enhance the precision of a collaborative filtering recommender system.

Our contributions are listed as follows:

- Recommender system with open linked data (RS-LOD) framework provides an interface to linked open data cloud that exploits the available data to solve the cold start problem in collaborative filtering.

- Matrix factorization (MF) model with LOD is introduced to handle the data sparsity problem in collaborative filtering. In the proposed approach, hidden feedback data and the proposed LOD semantic similarity measure (a combination of feature, distance, and statistical-based similarity methods) used as supplementary information to enhance the performance of recommendations.

- The proposed LOD-based RS framework is a generic model. It can be applied to any domain for recommendations. The key idea is to use the available semantic features in the LOD cloud for recommendations that will improve accuracy.

- Experiments are done on Netflix and MovieLens datasets to evaluate or to justify that proposed methods RS-LOD, MF with LOD and LOD similarity measures are better. From the observation, it is concluded that accuracy in recommendations improved for the proposed methods compared with previous methods.

The rest of the sections in this research article are arranged as follows. In Section 2, we discuss the related works, and then in Section 3, we explained the proposed methods like RS-LOD, MF-LOD, and LOD similarity. Section 4 represents the evaluation technique and result analysis. Finally, in Section 5 we conclude the work and a suggestion is given for future extension of this work.

2. Related Works

The data sparsity problem in collaborative filtering recommender system is addressed first, and then the cold start problem. So far, how unknown ratings are predicted, how new entities are handled, and what are the drawbacks in existing models were analyzed completely in this section. Assume X = {u_1, u_2, u_3, u_4, u_5, …., u_{233}, u_{234}, ….u_n} user set and Y = {i_1, i_2, i_3, .., i_{456}, i_{457}, …, i_m} item set. The relationship between items and users are represented in matrix form A = $(r_{u,i})_{n*m}$, where $r_{u,i}$ is the rating of user "u" to item "i" and "n" indicates the number of users, "m" indicated the number of items in the set respectively. For ratings, usually integer values from 1 to 5 are used in various datasets. $p_{u,i}$ is used to representing the predicted value for an unidentified rating.

2.1 Baseline Estimation

Initially, for unknown ratings researchers have taken the overall average rating μ, but practically that predicted rating is not fair for unrated users. Later, bias values for users and items are added in this calculation to improve prediction accuracy. For example, consider "*The Lion King"* movie; its overall average rating μ is 3.78. If the system puts 3.78 for unrated users, suppose he/she doesn't like an animation movie, then the entire prediction goes wrong. Now bias value is added for a particular user who says he/she doesn't like animation, and bias is -1 similarly for an

item that is also other than animation, no comedy, no proper narration, mean bias value -1. Now the predicted rating is (3.78- 1- 1) 1.78. So, now it's better when compared to the overall average rating μ. Estimation of unidentified rating as follows:

$$P_{u,i} = \mu + b_u(i) + b_t(j) \quad (1)$$

$b_t(j)$ and $b_u(i)$ are bias values of user "i" to item "j". Calculation of biases is implemented using least square problem optimization function as follows:

$$L = \min \boldsymbol{\Sigma}_{r_{u,i} \in A^+} (r_{u,i} - \mu - b_i - b_j)^2 + \lambda_u {b_u}^2 + \lambda_t {b_t}^2 \quad (2)$$

Where $\Sigma\, r_{u,i} \in {}_{A}{}^{+} (r_{u,i} - \mu - b_i - b_j)^2$ finds the bias of user b_i and item b_j that meets the known rating. The remaining term $\lambda_u {b_u}^2 + \lambda_t {b_t}^2$ used to avoid overfitting sparse data and A^+ is monitored ratings in the training dataset.

Later MF was introduced that relates many observable variables to a few latent variables. Various MF models were available, but singular value decomposition (SVD) is the most proficient method. SVD relates every user "i" to user latent vector "t_i" and every item "j" to item latent vector q_j. Rating matrix estimations are dependent on the product of user and item latent vector given in eq3.

$$P_{u,i} = \mu + b_u(i) + b_t(j) + {t_i}^T q_j \quad (3)$$

Using the following objective optimization function, calculate $b_u(i)$, $b_t(j)$, t_i and q_j.

$$L = \min \Sigma_{u,i \in A^+} (r_{u,i} - p_{u,i})^2 + \lambda_1 (||t_i||^2 + ||q_j||^2 + {b_u}^2 + {b_t}^2) \quad (4)$$

Where λ_1 (weight of regulation term) is used to avoid overfitting. This method is known as "biased SVD". From the analysis, it is discovered that existing MF based recommender systems not using semantic information and interactive data for unknown rating prediction and new entities.

2.2 LOD enabled RS

LOD forum asked users to publish their data in RDF format, which is used in LOD and connected different data sources to construct the LOD cloud that covers most of the knowledge and develops many data repositories such as DBpedia, LinkedMDB. Nowadays, LOD is

involved in many data-related applications, and RS is one that has benefited the most (Rajabi & Greller, 2019). Linked data semantic distance measurement is a similarity technique that is used to measure the semantic distance among different resources relying on the shortest path linking them in DBpedia. While calculating the distance between resources, it considers direct links, indirect links, incoming links, and outgoing links. However, the problem or limitation of this method is it ignores some similarity axioms like symmetry, minimality (Passant, 2012). Then, the Resim method was proposed to find the similarity between resources that satisfies all the axioms of similarity, but for the computation, it takes extensive time (Piao et al., 2015). Partitioned information content semantic similarity (PICSS) measures the level of semantic similarity among different resources based on the partitioned information content of their distinctive features. PICSS provides an organized procedure for precisely measuring similarity and promotes the semantic comparison and investigation of resources using linked data. PICSS is not scalable for recommendations (Meymandpour & Davis, 2016). Legato framework discovers unique links across RDF graphs in the web of open data. It addresses the dataset heterogeneity problem, predominantly those at the ontological level. Legato capacity is to avoid false positives by disambiguating successfully extremely similar instances across datasets with the help of a clustering method and ranking algorithm (Achichi et al., 2019). Linguistic linked open data (LLOD) publishes data for linguistics and natural language processing. The prime benefits of LLOD are representation, interoperability, and expressivity. (Chiarcos et al., 2018; Pico-Valencia et al., 2019). SemiLD framework combines linked data and heterogeneous semi-structured sources. It is an automated system that collects its input from various SPARQL endpoints and web APIs (Kettouch et al., 2019). SocialLink is a framework that automatically connects DBpedia entities to equivalent Twitter profiles (Nechaev et al., 2018). Framework XOSM was developed to integrate and query Open Street Map and Open Linked Geo Data resources. It contains a tool and XQuery library, which combines OSM layers and LGOD layers (Almendros-Jiménez et al., 2017). BROAD-RSI is an infrastructure designed to mine users' profiles and educational context from social media, which helps it to recommend educational resources (Pereira et al., 2018). The web contains datasets of heterogeneous nature and makes it difficult for users to query those datasets in order to exploit the vast amount of data. Different approaches evolved to overcome that limitation. A framework was created that allows end-users to acquire results from different datasets expressing the query using the vocabulary that the users are more familiar with and informs them about the quality of the answer. Moreover, this

framework serves technical users as a tool for establishing query rewriting benchmarks (Torre-Bastida et al., 2019). Volunteered geographic information (VGI), Semantic Web Interactive Gazetteer (SWI), and open linked data are used to include the missing geographic coordinates to biodiversity records (Cardoso et al., 2016). The linked open model enables the users to represent knowledge in the form of a diagram, a human-readable and data linkable model promoted by linked data (Karagiannis & Buchmann, 2016). Mobile technology in electronic learning allows learning at any moment. So, mobile and open linked data are used in a collaborative electronic learning environment. The knowledge base is built using open linked data techniques, leading to a situation of human learning by linking different data sources (Fermoso et al., 2015). A new personalized search method was proposed for the Web of Data depends on results categorization. In this method, search results are dynamically categorized into Upper Mapping and Binding Exchange Layer using a fuzzy retrieval method (Sah & Wade, 2016). Real-time passenger information is an important factor in making public transport easily accessible and more attractive to users. However, rural areas lack infrastructure, so it is difficult to provide information. With the help of open linked data, it is feasible to provide such services to users (Corsar et al., 2017). The DESIRE recommender system is designed to improve the performance of movie recommendations by keeping both the accuracy and diversity at an optimal level (Srinivasan & Mani, 2018). From the above analysis, it is concluded that none of the existing methods fully utilizes the available semantic information in LOD for cold start and data sparsity problem.

3. Proposed Approach: Recommender System with Linked Open Data

A new system is proposed called matrix factorization with open linked data (MF-LOD), which enhances the matrix factorization model based on implicit feedback data and linked open data based similarity measure to handle the data sparsity issue in collaborative filtering. On the other hand the recommender system with LOD (RS-LOD) model developed, which takes semantic features of items or users from the LOD cloud used to handle the cold start issue in recommendations.

3.1 LOD for cold start problem in CF(RS-LOD)

A cold start issue occurs in collaborative filtering because of the unavailability of data about that new entity. To solve this issue, it is advised to collect information from the open linked data

cloud “DBPedia” to handle efficiently the missing information. In architecture (figure 1), it is clearly mentioned that if a new entity (item or user) comes through the RS interface of the recommender system, it is sent to the regulator. Then it passes the information to query constructor, it will construct a SPARQL query to search missing data from the knowledge base through LOD interface, and it gives all information about that entity to the regulator. Information miner will filter and give the most important latent data. For example, given a movie name to the system, and it will give back its genre, actors, director, and also the available ratings (item metadata acquired from DBPedia). For new user, the proposed system retrieves demographic data and preferences from knowledge base LOD. Gathering such information is helpful to solve any new entity problem, and then we can recommend an item for a new user, or compute the similarity between new items and existing items in the database, from which a list of recommendations can be derived. The similarity calculator module will do the above said task and gives the data to regulator. Finally, the regulator sends the data to RS interface. Hence, in this way, cold start problem is resolved using LOD for recommender system (RS-LOD). Compared with sparse linear methods (SLIM,SSLIM) and PathRank, the proposed method RS-LOD performance is stable because it’s not only based on LOD knowledge base “DBPedia” (possibility of sparseness in semantic features) but also considers interaction information’s between user and item.

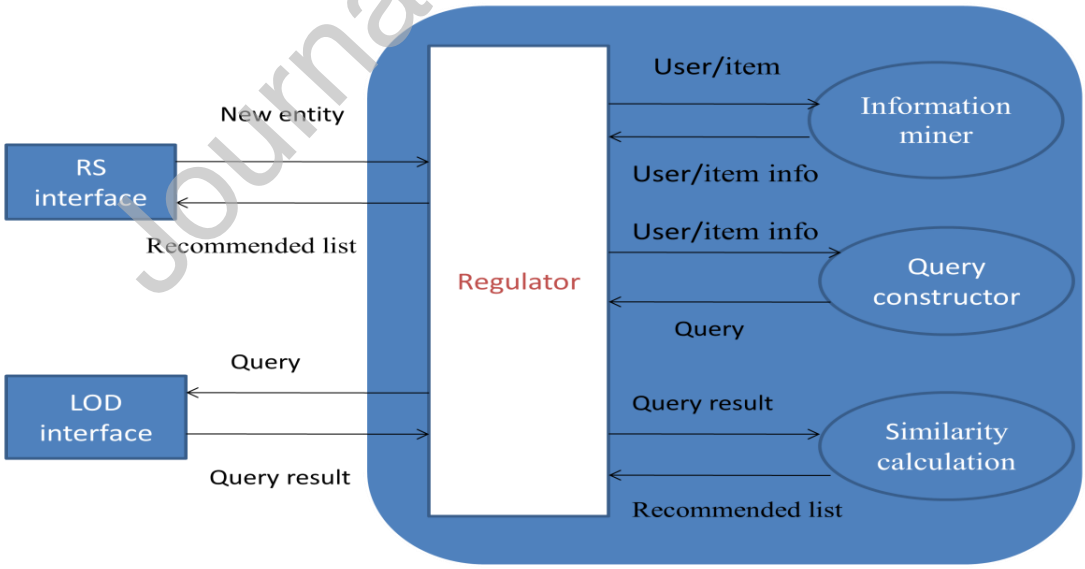


Figure 1: Architecture of the proposed RS-LOD system for cold start issue in CF

*3.2 Enhanced matrix factorization model for data spars*ity

The SVD++ model includes implicit feedback data to enhance user-factor vector in the matrix factorization process, resulting in better recommendations than with traditional SVD. In our approach, the same idea is used to enhance item-factor vector in MF as follows:

$$P_{u,i} = \mu + b_u(i) + b_t(j) + \left(q_j + |C(j)|^{-\frac{1}{2}} \Sigma_{u_h \in C(j)} X_h\right)^T * \left(t_i + |D(i)|^{-\frac{1}{2}} \Sigma_{t_l \in D(i)} Y_l\right) \quad (5)$$

Consider equation 5 here "μ" indicates the overall rating, then $b_u(i)$ and $b_t(j)$ are the bias values of user "i" and item "j". Where t_i and q_j are user and item vectors, C(j) is the set of users measured item "j". Here x_h is the input vector of user "h" to item modeling, y_l is the input vector of item l to user modeling. D(i) is the set of items evaluated by a particular user "i". This enhancement is done based on the assumption that there is some correlation between users' profiles and item profiles. This proposed method is called enhanced-SVD++.

3.3 Linked open data (LOD) similarity measure

Existing item similarity measures in collaborative filtering RS depend on a user-item rating matrix, which reduces the performance of recommendations on sparse datasets. To overcome a sparsity problem, LOD will help, as it connects all the related data from different domains using RDF format. A new semantic similarity measure is proposed that depends on LOD; it is a mixture of feature, distance, and statistical dependent metrics. PCC (Pearson correlation coefficient) is widely used for similarity measures in collaborative filtering, which identifies the correlation between two variables.

$$\text{Similarity}^{PCC}(p, q) = \frac{\sum_{i=1}^{n}(r_{ip} - r_{ap})(r_{iq} - r_{aq})}{\sqrt{\sum_{i=1}^{n}(r_{ip} - r_{ap})^2}\sqrt{\sum_{i=1}^{n}(r_{iq} - r_{aq})^2}} \quad (6)$$

Consider equation 6 where r_{ip} is the ranking of user "i" to item "p", similarly for $r_{iq.}$ r_{ap} average rating of item p and r_{ap} is the standard rating of item q, "n" is the number of users rated both items p and q. These types of users are called co-rating users. The drawback of PCC is that it doesn't consider the overlapping entries on similarity. See table1, showing that four users rated four items. Item 1 and item 3 have three co-rated users, and ratings are similar. However, PCC gives the result that item 1 and item 2 are more similar and have only two co-rated users.

Table 1: user-item rating matrix

	Item- 1	Item- 2	Item- 3	Item -4
User -1	3.5	3.4	3.2	-
User -2	2.7	-	2.5	-
User -3	-	-	-	2.9
User -4	1.8	1.8	1.75	4.2

To improve the rating similarity, minor modification is done on the Pearson correlation coefficient by adding a factor for a tiny amount of co-rating users as follows:

$$\text{Improvised}^{\text{PCC}}(p,q) = \max(0, \text{Similarity}^{\text{PCC}}(p,q).\frac{\text{Min}(|M(p)\cap M(q)|, \Delta)}{\Delta}) \quad (7)$$

Consider equation 7 where $\text{Similarity}^{\text{PCC}}$ (p,q) is traditional PCC, Δ is the penalty threshold of the amount of co-rated users. M(p) and M(q) are sets of users that evaluated item p, item q, and M(p)∩M(q) is the number of users that rated both items. Maximum range 0 to 1 is only considered because we take only positive similarity items. Still, limitations are there in similarity measure because correlations between semantic features of items are not considered. For further improvement in semantic similarity, we have to consider Partitioned Information Content Semantic Similarity (PICSS). PICSS is a combination of feature and information content based approach. It's an item based CF recommender model that builds on DBpedia. It considers all semantic features of item in DBpedia, which is time consuming. So, we propose a feature dependent semantic similarity measure $\text{sim}^{\text{PICSS}}$, depending on the most effective key features chosen by principal component analysis (PCA) because it's better than other feature selection techniques. Integration of improvised-PCC and modified partitioned information content semantic similarity $\text{sim}^{\text{PICSS}}$, improves the recommendation performance for sparse datasets. LOD semantic similarity measures consider specific features of items and also take similarity of ratings represented in equation 8.

$$\text{LOD sim}(p,q) = \text{Improvised}^{\text{PCC}}(p,q) + \text{sim}^{\text{PICSS}}(p,q) \quad (8)$$

3.4 Matrix factorization with hidden feedback and LOD similarity measure (MF-LOD)

Once semantic similarities between items are obtained, recommendations continue with the matrix factorization framework. Some extension is done on traditional matrix factorization to handle the data sparsity problem. The proposed system is shown in figure 2.

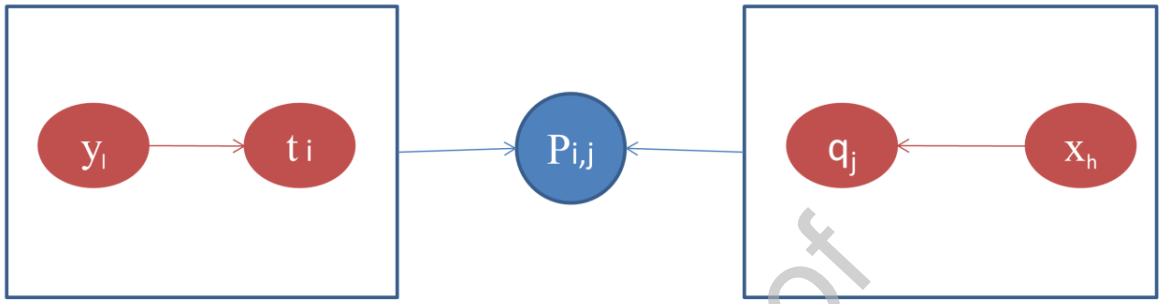


Figure 2: Linked open data – Matrix Factorization model

The mathematical formalization of the proposed system is given in detail as follows. User vector t_i is enhanced by hidden feedback data, and item vector q_j is enhanced by top k similar items. Forecasting is done by the following equation:

$$P_{u,i} = \mu + b_u(i) + b_t(j) + \left(q_j + |C^k(j)|^{-\frac{1}{2}} \Sigma_{v_h \in C^k(j)} X_h\right)^T * \left(t_i + |D(i)|^{-\frac{1}{2}} \Sigma_{u_l \in D(i)} Y_l\right) \quad (9)$$

Here $C^k(j)$ represents top "k" similar items related to item "j" are mined by improvised PCC and PICSS technique discussed in the LOD semantic similarity part. Where X_h indicates feature vector of i^{th} item in $C^k(j)$, defines the contribution of semantic similarity in item modeling and y_l indicates feature vector of i^{th} item in D(i), defines the contribution of hidden feedback data in user modeling. Where D(i) is various items rated by user "i". The specification in equation 9 are educated by resolving the optimization function, and it is formulated as regularized squared error of experimental ratings in training set as follows:

$$L = \min \sum_{r_{u,i} \in A^+} \left((r_{u,i} - p_{u,i})^2 + \lambda . \text{reg} \right) \quad (10)$$

Here "reg" represents regularization factor and "λ" indicates weight of that factor, and the same value is assigned to λ. The regularization factor is explained as follows:

$$reg = ||b_u(i)||^2 + ||b_t(j)||^2 + ||q_j||^2 + ||t_i||^2 + \Sigma_{h \in c^k(j)}||x_h||^2 + \Sigma_{l \in D(i)}||y_l||^2 \quad (11)$$

By checking the overall experiential ratings in the training set, the local minimum of the loss function is decided by using the SGD method. The gradient of $b_u(i)$, $b_t(j)$,q_j, t_i,,x_h, y_l are computed as follows.

For q_j, the derivative is computed as follows:

$$\frac{\partial L}{\partial q_j} = \frac{\partial L}{\partial (r_{u,i} - p_{u,i})} \cdot \frac{\partial (r_{u,i} - p_{u,i})}{\partial p_{u,i}} \cdot \frac{\partial p_{u,i}}{\partial q_j} + \lambda \cdot \frac{\partial reg}{\partial q_j} \quad (12)$$

Apply equations 9,10,11 and $e_{ij} = p_{u,i} - r_{u,i}$ be prediction error, equation 12 can be revised as follows:

$$\frac{\partial L}{\partial q_j} = 2 \cdot e_{ij} \cdot (t_i + |D(i)|^{-\frac{1}{2}} \Sigma_{u_l \in D(i)} y_l) + 2 \cdot \lambda \cdot q_j \quad (13)$$

Ignore constant co-efficient, we will obtain a gradient of q_j.

$$\frac{\partial L}{\partial q_j} = e_{ij} \cdot (t_i + |D(i)|^{-\frac{1}{2}} \Sigma_{u_l \in D(i)} y_l) + \lambda \cdot q_j \quad (14)$$

Similarly, the gradient for t_i is:

$$\frac{\partial L}{\partial t_i} = e_{ij} \cdot (q_j + |C^k(j)|^{-\frac{1}{2}} \Sigma_{v_h \in C^k(j)} x_h) + \lambda \cdot t_i \quad (15)$$

Similarly, the gradient for x_h and y_l according to equation 9 is:

$$\frac{\partial L}{\partial x_h} = e_{ij} \cdot |C^k(j)|^{-\frac{1}{2}} (t_i + |D(i)|^{-\frac{1}{2}} \Sigma_{u_l \in D(i)} y_l) + \lambda \cdot x_h \quad (16)$$

$$\frac{\partial L}{\partial y_l} = e_{ij} \cdot |D(i)|^{-\frac{1}{2}} (q_j + |C^k(j)|^{-\frac{1}{2}} \Sigma_{v_h \in C^k(j)} x_h) + \lambda \cdot y_l \quad (17)$$

Finally, the gradient of $b_u(i)$ and $b_t(j)$ as follows:

$$\frac{\partial L}{\partial b_u(i)} = e_{ij} + \lambda \cdot b_u(i) \quad (18)$$

$$\frac{\partial L}{\partial b_t(j)} = e_{ij} + \lambda \cdot b_t(j) \quad (19)$$

The process of stochastic gradient descent (SGD) algorithm for linked open data based MF model is explained below. Here "n" is the learning rate.

Algorithm *SGD algorithm for MF-LOD model*

Input: C, D, k, n, λ and A

Output: latent factor t,q,x,y, b_u, and b_t

Initialize: "t" and "q" with some random variable sampling taken from Gaussian distribution (GD) zero mean and set zeros to variance x,y, b_u, and b_t.

While "L" don't coverage **Do**

Calculate gradients using equations 14, 15, 16, 17, 18, and 19, and update latent factors as follows:

$t_i = t_i - n \cdot \partial L/\partial t_i$, where i from 1,2, ...m

$q_j = q_j - n \cdot \partial L/\partial q_j$, where j from 1,2, ...n

$x_h = x_h - n \cdot \partial L/\partial x_h$, for all $v_h \in C^k(j)$

$y_l = y_l - n \cdot \partial L/\partial y_l$, for all $u_l \in D(i)$

$b_u(i) = b_u(i) - n \cdot \partial L/\partial b_u(i)$ where i from 1,2...m

$b_t(j) = b_t(j) - n \cdot \partial L/\partial b_t(j)$ where j from 1,2,3,4...n

End while

Return t, q, x, y, b_u, b_t

Hence, open linked data resolves the data sparsity and cold start issue in CF recommender system.

4. Experimental Analysis and Results

All Testing, implementations and comparisons are done in a similar experimental environment, 1.6 GHz Intel Core i5 processor with 8GB ram.

4.1 Data set

For rating predictions, two different data sets are used in our experiments, MovieLens, and Netflix. The MovieLens-20M dataset contains 27,000 movies and 138,000 users with 20 million

ratings, the sparsity of its rating matrix is 99.46%. Netflix-20M dataset contains 4,499 movies and 470,758 users with 24,053,764 ratings; sparsity of its rating matrix is 98.86%.

Sparsity = 100 - (available-ratings / (number of users*number of items))*100 (20)

Equation 20 is used to calculate the sparseness in the dataset. Sparsity means a lot of information is missing or most of the entries are zero in rating matrix. The mapping between MovieLens and Netflix items with the DBpedia knowledge base is available, which is the key source for the research work displayed in table 2.

Table 2: Sample mappings between Movielens and Netflix items with DBPedia URI

Item-id	Item-Name	DBPedia URI
889	Lion King (1994)	http://dbpedia.org/resource/lion_king_(film)
1258	The Mummy (1997)	http://dbpedia.org/resource/Mummy(Movie)
13589	Angry birds (2016)	http://dbpedia.org/resource/angry_birds
20159	Annabelle comes home (2019)	http://dbpedia.org/resource/Annabelle3_comeshome

Table 3: Datasets and its Characteristics

	Users	Items	Ratings	Sparsity
MovieLens 20M	138000	27000	20-M	99.46%
MovieLens-DBPedia 20M	138000	25589	18.6 M	99.5%
NetFlix 20M	470758	4499	24053764	98.86%
NetFlix-DBPedia 20M	470758	4038	20068125	98.94%

This mapping will reduce the sparsity problem of these datasets. In some situations, items that exist in MovieLens and Netflix have no corresponding entries in DBpedia. Table 3 represents the characteristics of the datasets used for experimental evaluation.

By applying the PCA method, we extracted the most effective semantic features, including subject, director, genre, and stars of movies. Not every item in MovieLens and Netflix has such absolute feature mapping with DBpedia. Sparsity is there in both datasets mapped with DBpedia for the selected features. First, we split datasets into two parts, with 30% of entire ratings as a test set and the remaining as a training set. Later, we apply five-fold cross-validation by choosing a test and training sets randomly. For the cold start issue the same datasets were used, and the DBpedia knowledge base is utilized to gather information about new user/item.

4.2 Evaluation metrics and methods

To demonstrate the effectiveness of the proposed linked open data based matrix factorization method, it is compared with the existing methods like biased singular value decomposition, SVD++, recommender system with PICSS and SVD. Finally, LOD similarity measure the proposed method is compared with the existing similarity methods by applying them in the matrix factorization model. The following metrics are used here to assess the performance of various recommender methods: Precision, Recall, F1-score, Mean Absolute Error, and Root Mean Square Error.

4.3 Experimental results

In this section, proposed MF-LOD is compared with other recommender methods. RS-LOD, enhanced SVD++, and LOD similarity measure are also compared with existing techniques. The outcome is the proposed method outperforms the existing methods.

4.3.1 Comparing various recommender methods

Ratings used in our experiment range from 1 to 5. While computing recall and precision of a recommendation list, items with a rating of more than 4 are considered as likes. The performance analysis of various recommender methods for MovieLens and Netflix datasets is shown in table 4 and table 5.

Table 4: Performance of various methods on Netflix 20M-dataset

	MAE	RMSE	Precision@40	Recall@40	F1-Score@40
Baseline (SVD)	0.705	0.851	0.531	0.558	0.547
Biased Singular Value Decomposition	0.687	0.836	0.541	0.563	0.549
Singular Value Decomposition ++	0.650	0.819	0.547	0.566	0.554
Open Linked Data based Matrix Factorization	0.613	0.801	0.556	0.571	0.560

Table 5: Performance of various techniques on MovieLens 20M-dataset

	MAE	RMSE	Precision@25	Recall@25	F1-Score@25
Baseline (SVD)	0.724	0.885	0.498	0.538	0.535
Biased Singular Value Decomposition	0.708	0.851	0.522	0.545	0.544
Singular Value Decomposition ++	0.689	0.837	0.518	0.542	0.541
Open Linked Data based Matrix Factorization	0.670	0.811	0.564	0.560	0.556

From the tables, it is understood that no matter which method is used, the results on Netflix are better than MovieLens, which indicates that data sparsity has a major impact on recommendation performance. Biased singular value decomposition is better than the baseline method called SVD (discussed in section 2.1) because of the utilization of matrix factorization in CF. Then SVD++ is good when compared to biased-SVD; this is due to the inclusion of hidden feedback data in CF. The proposed open linked data based matrix factorization is better than other methods due to the extension of the other side information from the LOD cloud in CF and MF process. Our approach achieves better recommendation performance than other models on both datasets. Precision, Recall, and F1-score have similar trends like MAE and RMSE. An increase in the

length of the recommendations list decreases the precision value gradually. F1-score touches peak value when the recommendations list length is 40 for Netflix, and for MovieLens it's 25. Therefore, while comparing precision and recall of various techniques N value of Top-N, surely it will vary on these two datasets.

4.3.2 Effect of latent factors, neighbor size, and penalty threshold

Let X is the number of latent factors of user and item vector in the matrix factorization framework. If you take diverse X value, it will produce various recommendation accuracy. So, we analyzed the impact of X on the performance of MF-LOD based recommender models. From the table, it is observed that RMSE decreases as X increases for all methods. Experimental results show that the proposed enhanced-SVD++ model improves the accuracy of recommendations for different latent factors. See table 6.

Table 6: Prediction accuracy is calculated by RMSE on MovieLens test sets for a diverse number of factors

Model	50-factors	100-factors	200-factors
Singular Value Decomposition	0.904	0.902	0.900
Singular Value Decomposition ++	0.895	0.892	0.891
Enhanced- Singular Value Decomposition ++	0.866	0.852	0.841

It is simple to understand that the number of neighbors increases the similarity between the item and its neighbors will automatically decrease. If the penalty value is too small, there is no effect on recommendations performance, but it reaches maximum (100), then definitely there is an improvement or convergence in recommendation accuracy.

4.3.3 Evaluation result for cold start issue

To determine the effectiveness of the proposed RS-LOD method for a cold start issue, MAE is calculated for the Netflix dataset shown in table 7, and it is concluded that the proposed method RS-LOD with LOD similarity has a lower error rate than other existing methods with Pearson similarity. The system calculates the similarity between the new and available entity in the database using the formula given below:

SIM (NE, EE) = $(\sum_{i=1}^{n} simfea_i)$/n (21)

$simfea_i$ = (f_i NE ∩ f_i EE) / (f_i NE ∪ f_i EE) (22)

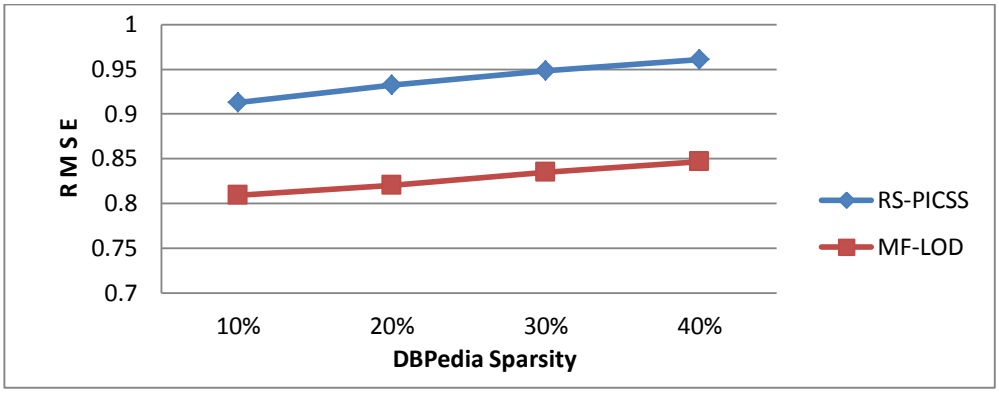
Here NE is a new entity and EE is an existing entity in the database. Simfea is similarity based on specific entity feature "i" (entity represents user or item). SIM is a global similarity between the new entity and an existing entity. Here "n" is the total number of features gathered from LOD. f_i NE, f_i EE is a feature of new entity and feature of another entity respectively. Mean Absolute Error (MAE) is calculated as predicted note of user "u" on the item "i" subtracted with note already given by the user on the item. Summation of entire value divided by "n"(the total number of predicted notes) gives you the MAE.

Table 7: Comparative results

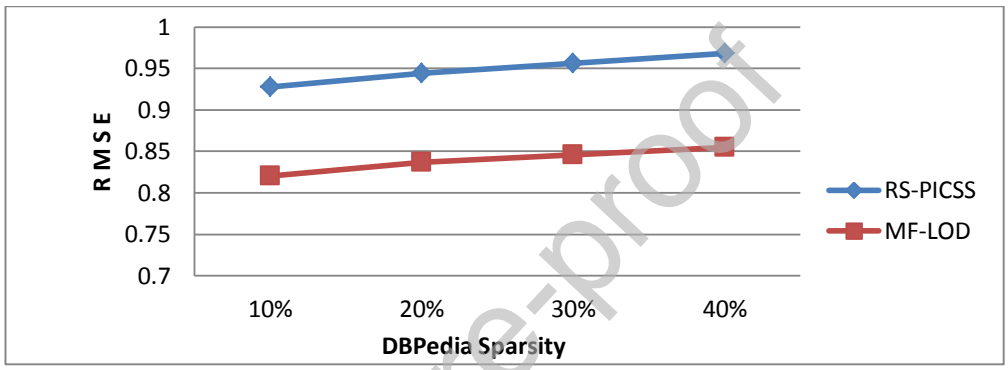
Method	MAE
CF_U	0.86
CF_I	0.83
RS-LOD	0.71

4.3.4 Issue on feature sparsity in the knowledge base

To assess the feature sparsity in the knowledgebase (DBpedia) on recommendations, we trim MovieLens and Netflix feature mappings with DBpedia manually at various scale "y" (10%,20%) and we keep the original data with a probability of 1- y to imitate diverse sparsity of semantic features. Figure 3 represents the performance of the recommender system with PICSS and MF-LOD the proposed method in various feature sparsity on two datasets. From the figure, it is understood that feature sparsity impacts the recommendation performance, as the feature decreases RS performance also declining. Compared to a recommender system with partitioned information content semantic similarity, MF-LOD is better because the item similarity measure is based on the LOD knowledge base. Enhanced data quality caused by using LOD (multiple data sources) can drastically improve the recommendation performance.



(a)

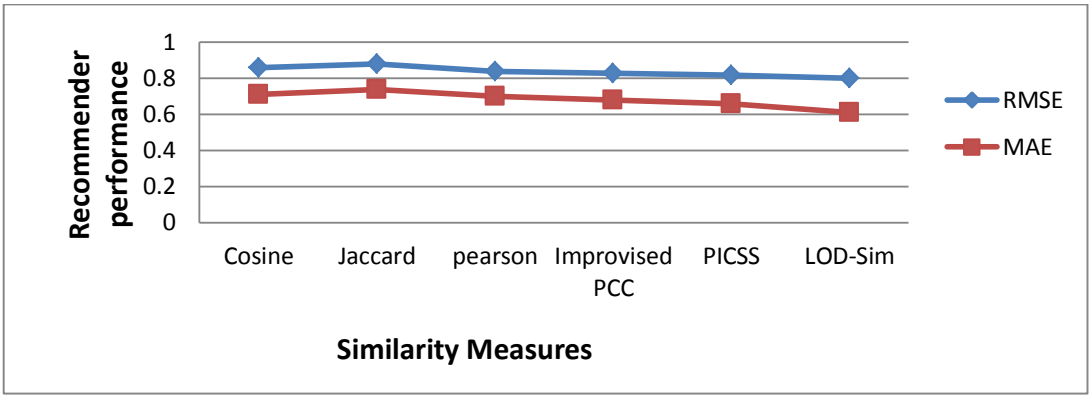


(b)

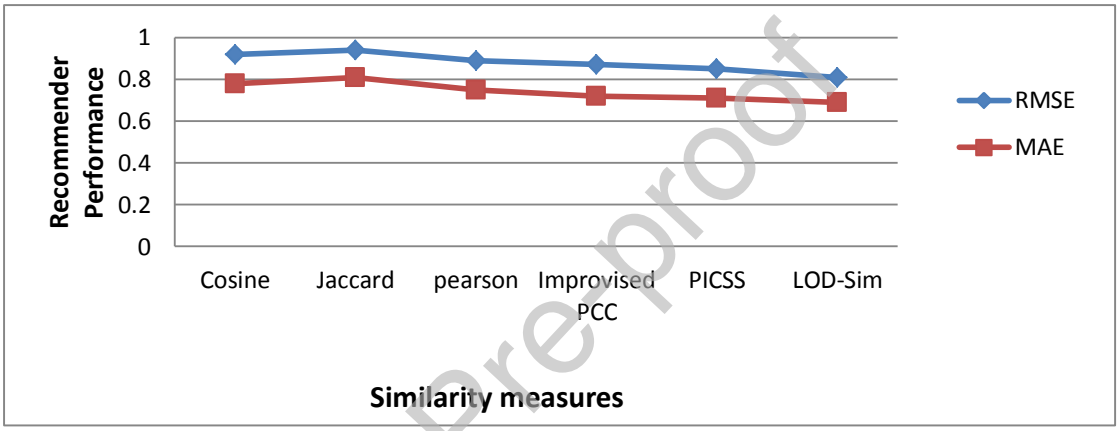
Figure 3: Comparison of feature sparsity in Knowledgebase based on RMSE on (a) Netflix and (b) Movielens datasets

4.3.5 Comparison of various similarity measures

Here the performance of the proposed LOD similarity measure is compared with the existing similarity measures. Figure 4 exhibits the performance of various similarity measures. It is understood from the figure that the Pearson method is better than Jaccard and cosine. Improvised PCC is better than Pearson with the inclusion of a penalty factor in similarity measures, which improves recommendation performance. PICSS outperforms other methods with the inclusion of shared features of item pairs and can extensively improve recommendation performance. Finally, the proposed LOD similarity achieves better accuracy than other methods by considering semantic features from the LOD knowledge base.



(a)



(b)

Figure 4: Comparison of different similarity measures on (a) Netflix dataset (b) MovieLens dataset

5. Conclusion

In this research article, we focused on cold start problem and data sparsity issue in collaborative filtering method. Recommendation accuracy is improved with the integration of linked open data with collaborative filtering. Extension work is done on matrix factorization model used in collaborative filtering RS to deal with the data sparsity issue. Using the LOD knowledge base, the cold start issue is solved by gathering information about that new entity. We analyzed all computation techniques used in existing systems and identified that feature dependent data helps to develop the efficiency of recommender systems. A new similarity measure is introduced here called LOD similarity, which is a combination of improvised PCC + similarity PICSS that is used to discover semantically similar items of a target item to expand item-factor vector in the matrix factorization process to improve accuracy. Based on linked open data similarity measure,

a linked open data-matrix factorization model is proposed. In MF-LOD, an extension is done on the user side as well as the item side by adding hidden feedback data and semantic features of items from DBpedia. Our work is a generic model, so researchers can apply this for different knowledge bases to relieve a data sparsity problem. We also described how LOD could resolve a new entity problem in CF-RS. The cause of this problem is information lacking, and LOD can cover it. The comparison experiment results show that the proposed improvised similarity method achieves improved performance in rating forecast than existing models. Linked open data matrix factorization performs better when compared to other MF models in a data sparsity issue. Semantic features available in the open linked data cloud play a key role in RS, and the data is utilized to handle data sparsity problem and cold start issue. In this proposed system, only DBpedia is used to acquire additional information. However, in the future, LOD knowledge bases like Freebase, LinkedMDB, and YAGO are used to mine more constructive information that will increase the performance of recommender systems. In addition, users' social relationships in online social networks are also used as additional information in the matrix factorization process to handle data sparsity and the cold start issue. In the future, incorporating Deep Learning with this system will produce better results.

Credit authorship contribution statement

Senthilselvan N: Software, Data curation, Writing - original draft. Subramaniyaswamy V: Conceptualization, Methodology, Supervision. Amir H. Gandomi: Writing - review & editing, Supervision. Sivaramakrishnan N: Validation, Visualization, Investigation.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Achichi, M., Bellahsene, Z., Ellefi, M. B., & Todorov, K. (2019). Linking and disambiguating entities across heterogeneous RDF graphs. *Journal of Web Semantics*, *55*, 108-121.

Almendros-Jiménez, J. M., Becerra-Terón, A., & Torres, M. (2017). Integrating and Querying OpenStreetMap and Linked Geo Open Data. *The Computer Journal*, *62*(3), 321-345.

Billsus, D., & Pazzani, M. J. (1998, July). Learning Collaborative Information Filters. In *Icml* (Vol. 98, pp. 46-54).

Cardoso, S. D., Amanqui, F. K., Serique, K. J., dos Santos, J. L., & Moreira, D. A. (2016). SWI: a semantic web interactive gazetteer to support linked open data. *Future Generation Computer Systems*, *54*, 389-398.

Chiarcos, C., Khait, I., Pagé-Perron, É., Schenk, N., Fäth, C., Steuer, J., ... & Wang, J. (2018). Annotating a Low-Resource Language with LLOD Technology: Sumerian Morphology and Syntax. *Information*, *9*(11), 290.

Corsar, D., Edwards, P., Nelson, J., Baillie, C., Papangelis, K., & Velaga, N. (2017). Linking open data and the crowd for real-time passenger information. *Journal of Web Semantics*, *43*, 18-24.

Cui, L., Huang, W., Yan, Q., Yu, F. R., Wen, Z., & Lu, N. (2018). A novel context-aware recommendation algorithm with two-level SVD in social networks. *Future Generation Computer Systems*, *86*, 1459-1470.

Del Corso, G. M., & Romani, F. (2019). Adaptive nonnegative matrix factorization and measure comparisons for recommender systems. *Applied Mathematics and Computation*, *354*, 164-179.

Di Noia, T., Ostuni, V. C., Rosati, J., Tomeo, P., Di Sciascio, E., Mirizzi, R., & Bartolini, C. (2016). Building a relatedness graph from linked open data: A case study in the IT domain. *Expert Systems with Applications*, *44*, 354-366.

Fermoso, A. M., Mateos, M., Beato, M. E., & Berjón, R. (2015). Open linked data and mobile devices as e-tourism tools. A practical approach to collaborative e-learning. *Computers in Human Behavior*, *51*, 618-626.

Hsieh, M. Y., Chou, W. K., & Li, K. C. (2017). Building a mobile movie recommendation service by user rating and APP usage with linked data on Hadoop. *Multimedia Tools and Applications*, *76*(3), 3383-3401.

Karagiannis, D., & Buchmann, R. A. (2016). Linked open models: extending linked open data with conceptual model information. *Information Systems*, *56*, 174-197.

Kettouch, M., Luca, C., & Hobbs, M. (2019). SemiLD: mediator-based framework for keyword search over semi-structured and linked data. *Journal of Intelligent Information Systems*, *52*(2), 311-335.

Lnenicka, M., & Komarkova, J. (2019). Developing a government enterprise architecture framework to support the requirements of big and open linked data with the use of cloud computing. *International Journal of Information Management*, *46*, 124-141.

Luo, L., Xie, H., Rao, Y., & Wang, F. L. (2019). Personalized recommendation by matrix co-factorization with tags and time information. *Expert Systems with Applications*, *119*, 311-321.

Meymandpour, R., & Davis, J. G. (2016). A semantic similarity measure for linked data: An information content-based approach. *Knowledge-Based Systems*, *109*, 276-293.

Najafabadi, M. K., Mohamed, A., & Onn, C. W. (2019). An impact of time and item influencer in collaborative filtering recommendations using graph-based model. *Information Processing & Management*, *56*(3), 526-540.

Nechaev, Y., Corcoglioniti, F., & Giuliano, C. (2018). SocialLink: exploiting graph embeddings to link DBpedia entities to Twitter profiles. *Progress in Artificial Intelligence*, *7*(4), 251-272.

Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications*, *92*, 507-520.

Oliveira, J., Delgado, C., & Assaife, A. C. (2017). A recommendation approach for consuming linked open data. *Expert Systems with Applications*, *72*, 407-420.

Passant, A. (2010, November). dbrec—music recommendations using DBpedia. In *International Semantic Web Conference* (pp. 209-224). Springer, Berlin, Heidelberg.

Pereira, C. K., Campos, F., Ströele, V., David, J. M. N., & Braga, R. (2018). BROAD-RSI–educational recommender system using social networks interactions and linked data. *Journal of Internet Services and Applications*, *9*(1), 7.

Piao, G., showkat Ara, S., & Breslin, J. G. (2015, November). Computing the semantic similarity of resources in dbpedia for recommendation purposes. In *Joint International Semantic Technology Conference* (pp. 185-200). Springer, Cham.

Pico-Valencia, P., Holgado-Terriza, J. A., & Senso, J. A. (2019). Towards an Internet of Agents model based on Linked Open Data approach. *Autonomous Agents and Multi-Agent Systems*, *33*(1-2), 84-131.

Rajabi, E., & Greller, W. (2019). Exposing Social Data as Linked Data in Education. *International Journal on Semantic Web and Information Systems (IJSWIS)*, *15*(2), 92-106.

Rowe, M. (2014, August). SemanticSVD++: incorporating semantic taste evolution for predicting ratings. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* (Vol. 1, pp. 213-220). IEEE.

Sah, M., & Wade, V. (2016). Personalized concept-based search on the Linked Open Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, *36*, 32-57.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). *Application of dimensionality reduction in recommender system-a case study* (No. TR-00-043). Minnesota Univ Minneapolis Dept of Computer Science.

Srinivasan, U. S., & Mani, C. (2018).Diversity-Ensured Semantic Movie Recommendation by Applying Linked Open Data.

Torre-Bastida, A. I., Bermúdez, J., & Illarramendi, A. (2019). Estimating query rewriting quality over LOD. *Semantic Web*, (Preprint), 1-26.

Vozalis, M. G., & Margaritis, K. G. (2007). Using SVD and demographic data for the enhancement of generalized collaborative filtering. *Information Sciences*, *177*(15), 3017-3037.

Wang, R., Cheng, H. K., Jiang, Y., & Lou, J. (2019). A novel matrix factorization model for recommendation with LOD-based semantic similarity measure. *Expert Systems with Applications*, *123*, 70-81.

Yuan, X., Han, L., Qian, S., Xu, G., & Yan, H. (2019). Singular value decomposition based recommendation using imputed data. *Knowledge-Based Systems*, *163*, 485-494.