2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20), 14-17 December 2020, Melbourne, Australia.

Using Social Tag Embedding in a Collaborative Filtering Approach for Recommender Systems

Diego Sánchez-Moreno, María N. Moreno-García
University of Salamanca
Salamanca, Spain
sanchez91@usal.es, mmg@usal.es

Nasim Sonboli
University of Colorado Boulder
Boulder, Colorado, USA
nasim.sonboli@colorado.edu

Bamshad Mobasher
DePaul University
Chicago, Illinois, USA
mobasher@cs.depaul.edu

Robin Burke
University of Colorado Boulder
Boulder, Colorado, USA
robin.burke@colorado.edu

***Abstract*— Nowadays, the use of social information is extending to more and more application domains. In the field of recommender systems, this information has been exploited in different ways to address some problems, especially associated with collaborative filtering methods, and thus achieve more reliable recommendations. Specifically, social tagging is used in this area mainly to characterize the items that are the subject of the recommendations. In this work, a user-based collaborative filtering approach is presented, where tags processed by word embedding techniques are used to characterize users. User similarities based on both tag embedding and ratings are combined to generate the recommendations. In the study conducted on two popular datasets, the reliability of this approach for rating prediction and top-N recommendations was tested, showing the best performance against the most widely used collaborative filtering methods.**

Keywords-word embedding; social tagging; recommender systems; collaborative filtering

I. Introduction

Collaborative Filtering (CF) methods traditionally used in recommender systems use only information about preferences in the form of ratings that users provide either explicitly or implicitly. These approaches generally perform well under ideal conditions, when the number of ratings given by users to items is high enough and their distribution for users and items is uniform. However, since this situation is uncommon in real systems, these methods suffer a series of well-known problems such as sparsity, cold-start, grey-sheep, or popularity bias, among others [1-4]. To address these drawbacks of CF methods, different approaches have emerged, many of which make use of user or item attributes to supplement the rating data. Generally, user demographic information (age, occupation...) or item characteristics (genre of a movie, rhythm of a song...) are used as attributes. Current availability of user social information, through the own recommender systems or other social networks, has allowed to take advantage of it to deal with these problems.

Tags are among the most used types of social information in recommender systems. Unlike other types of labelling, social tagging is not restricted to a set of predefined tags and are not used for classification into taxonomies, but tags that social network users associate with items contain free text and form a broad domain-specific body of knowledge that is called folksonomy [5]. This is an informal way of indexation that does not stablish a hierarchical organization or any other relationship between tags. Despite this lack of organization, social tagging allows the creation of a dynamic and very rich user-driven description of items in multiple dimensions. However, its use is not exempt from problems such as redundancy or ambiguity, among others. Therefore, to overcome those drawbacks, a pre-processing of the tags is required, which will depend on their subsequent use.

Word embedding techniques are being used to address some of the problems that arise when using tags in recommender systems. These methods treat the words in a sentence considering their context, which is formed by the surrounding words. Social tags that are part of folksonomies can be processed in an analogous way to words, and thus minimize problems of ambiguity and inconsistency by considering the context. The basics of word embedding involving the learning of distributed representations of words through neural networks, were first introduced by Bengio et al. [6]. This approach becomes popular when Mikolov et al. proposed two model architectures for computing continuous vector representations of words: Continuous Bag-of-Words (CBOW) and Skip-gram [7]. These two models are now being widely applied in many areas.

Although social tags have been exploited for some time in recommender systems in different proposals in the literature, much less work has been done using tag embedding. In addition, in most proposals, tags are used as item attributes to classify or characterize them. In this work, social tags that users assign to items are processed by means of word embedding techniques in order to use them to characterize users. The main contribution of this work is the proposal of a user-based CF method that uses the results of tag embedding in a complementary way to the rating data to establish the similarity between users. Therefore, the recommendations provided by this approach are based on the affinity between users both in terms of their preferences for items, and in terms of the information underlying the tags they assign to them. In this way, the reliability of the recommendations is improved.

II. RELATED WORK

Social tagging has been used for several years in different application domains for purposes such as search personalization, categorization and recommendation of resources. In the field of recommender systems, many of the proposals are aimed at recommending the tags themselves [8] [9], while others use social tagging as implicit feedback to recommend other items. There are far fewer papers in which tags are used as complementary information to ratings obtained either explicitly or implicitly.

Social tags are useful to determine the similarity between items and make content-based or hybrid recommendations [10], sometimes addressed to solve some particular problems of CF methods such as cold-start [11]. Tags and other social information have been combined in some recommender system proposals as the presented in [12] in the field of music where the detection of communities in social networks and the creation of sub-clustering from tags of artists are the basis of this approach. In [13], user preferences for specific types of music are obtained from top tags for tracks and artists. Then, a conventional CF method is applied to compute tag-based user similarities and make recommendations. In [14], TF-IDF obtained from social tags that users give to music items is used to infer user expertise and incorporate it as a weight in the calculation of recommendations. Neural networks have also been used in tag-based recommender systems to extract the in-depth features from tags representing users' profiles [15, 16].

As stated before, social tags are not organized in a hierarchical classification as taxonomies but form a wide body of knowledge known as folksonomy that allows for multidimensional classification. Although folksonomies provide a richer description of the resources, they have the disadvantage of being noisy due to the presence of ambiguity and redundancy in the tags. Two main approaches have been proposed to address this problem, latent semantic analysis [17-19], and clustering [20, 21]. Noise removal on tags is also addressed in [22] by finding relevant tags for each item with their corresponding weights and tag preferences for each user from the multiple relationships among users, items and tags. Other drawback of tag-aware recommender systems is the sparsity, since not all users assign tags to the items. The proposal presented in Zhang et al. [23] is focused on capturing the transitive associations between users and items with tags to minimize this problem. Sparsity and high dimensionality of tagging information are addressed in [24] through a three-step process based on matrix factorization that includes a ranking-oriented optimization model, an explicit-to-implicit feature mapping scheme, and the regularization of user latent features using users' neighbor relationships.

More recently, word embedding has also been used to deal with tag problems in recommender systems especially in content-based approaches. Ambiguities and inconsistencies are minimized because this technique does not treat tags in isolation but considers their context. In [25], a tag embedding based approach is proposed to predict links in social networks as well as to recommend items through a neural binary classifier that receives as input tag embedding of users and items. The results show the robustness of the method even in sparse contexts where there are few user-item interactions. However, this proposal does not allow to predict the value of the rating that a given user would assign to a given item, but simply whether an item should be recommended or not. Therefore, only rank-based metrics are used in the validation.

Liang et al. [26] proposed a tag-aware recommendation method for rating prediction that simulates matrix factorization through learning latent features for users and items jointly using two different neural networks. In this proposal, item metadata are used in addition to tags. The validation with a movieLens dataset shows a significant improvement over other CF methods, although results are not provided for top-N recommendations.

A proposal for top-N recommendations, the tag-aware neural attention model (TNAM) distributed in a four-layer architecture, is presented in [27]. In the input layer, sparse data about items, user and tags are transformed in four embedding vectors (user-tag, item-tag, user and items), which pass to the next layer where correlations among users, items and tags are learned by means of a neural CF method. In the following layer, different weights for different users are assigned to the same tag of an item to capture user-item interests. At the top, a general multi-layer perceptron network generates the recommendation lists. This approach is validated with three public datasets, from which only the records containing user-tag interactions are retained. In addition, user preferences are inferred only from the annotation of users on items, without considering any other interaction or explicit user feedback.

The approach proposed in this paper is quite different from those based on tag embedding described above. On the one hand, the only data required are tags and implicit or explicit ratings. No additional information about users or items is required. On the other hand, tags are used as a complement to ratings to improve recommendations, unlike other work where tags are used as the only feedback to derive user preferences. In addition, the output is not binary as in other methods, but is a real numerical value corresponding to the predicted rating. However, the method is not only suitable for rating prediction but also for top-N recommendations, so the validation has been done for both purposes with the specific metrics of each.

III. TAG EMBEDDING-BASED CF APPROACH

The recommendation method presented in this paper is an extension of the user-based collaborative filtering approach in which the information from social tags given to items by users is exploited. That information is extracted by means of word embedding techniques and incorporated to the CF model.

A. User-Based Collaborative Filtering

User-based CF techniques base the recommendations on the similarity between users in terms of item preferences. Consequently, a given user will be recommended items that similar users have rated positively.

Given a set of m users $U = \{u_1, u_2, \ldots u_m\}$ and a set of n items $I = \{i_1, i_2, \ldots i_n\}$, each user u_i have a list of ratings given to a set of items I_{u_i}, where $I_{u_i} \subseteq I$. These ratings are represented by the matrix of ratings $\mathbf{R} \coloneqq r_{ij}$ where $\mathbf{R} \in M_{n \times m}(\mathbb{N})$.

Then, a recommendation for the active user $u_a \in U$ involves a set of items $I_{ra} \subset I$ that fulfill the condition $I_{ra} \cap I_{ua} = \varnothing$, since only items not rated by u_a can be recommended. The similarity between users is computed from ratings by using some metrics such as cosine similarity, which is one of the most widely used. Similarity between the active user u_a and another user u_i is denoted as $sim(u_a, u_i)$. Cosine similarity for two given users, u_a and u_i, is computed according to eq. 1, where r_{aj} and r_{ij} are the ratings of user u_a and user u_i for item i_j respectively, and V_{u_a} and V_{u_i} are the vectors containing the ratings given to items by users u_a and u_i respectively.

$$sim(u_a, u_i) = cos\left(V_{u_a}, V_{u_i}\right) = \frac{\sum_{j=1}^{n} r_{aj}\, r_{ij}}{\sqrt{\sum_{j=1}^{n} r_{aj}^2}\sqrt{\sum_{j=1}^{n} r_{ij}^2}} \quad (1)$$

The ratings of the most similar users, the k nearest neighbors, are used to predict the rating that the active user would give to an item i_j that he/she has not played yet, by means of eq. 2.

$$pr_{aj} = \bar{r}_a + \frac{\sum_{i=1}^{k} sim(u_a,u_i)(r_{ij} - \bar{r}_i)}{\sum_{i=1}^{k} |sim(u_a,u_i)|} \quad (2)$$

In the approach proposed in this work, not only similarity based on ratings but also similarity based on social tags is used. To compute this similarity, the word embedding and PCA (Principal Component Analysis) techniques are applied to the tags as described below.

B. Social tag embedding

Word embedding techniques are widely used in NLP (Natural Language Processing) for numerical representation of words in documents where vectors of words are learned by neural networks. These representations allow to capture the word context unlike other kinds of representations using different frequency functions. Two popular word embedding model architectures for learning distributed representations of words are Continuous Bag-of-Words (CBOW) and Skip-gram [7]. These are unsupervised models that try to minimize computational complexity, although they can be applied to a vast corpus of words to create a vocabulary and generate dense word embeddings for each word.

CBOW is used to predict the probability of a word given a context, which may be a single word or a group of words. The objective function in CBOW is negative log likelihood of a word given a context:

$$-\log(p(w_o|w_i)) \quad (3)$$

The softmax function is used for computing the probability:

$$p(w_o|w_i) = \frac{\exp({v'_{w_o}}^T v_{w_i})}{\sum_{w_i=1}^{W} \exp({v'_{w_o}}^T v_{w_i})} \quad (4)$$

Where v_{w_i} and v'_{w_o} are the input and output vector representations of w, and W is the number of words in the vocabulary.

In the CBOW model, the distributed representations of context are combined to predict the word in the middle.

The Skip-gram model is similar to CBOW, although its purpose is to predict the context given the current word. The calculations up to hidden layer activations are the same but the input in this case is the current word.

These models can be applied to social tags instead of words. However, the way in which this is done depends on the objective to be achieved, since the tags are not contained in sentences as is the case with the words. We propose to obtain feedback from the user's social tagging on the items in a manner analogous to that obtained from the ratings. Therefore, in our proposal words are replaced by tags and sentences will be formed by the tags corresponding to each user-item pair, in the same way that the ratings are associated with those pairs.

There are other deep learning-based approaches for word embedding that could have been used. However, previous studies with different datasets have shown that the improvements achieved, at the expense of much higher computational cost [28], are not very significant and depend largely on the dataset on which they are applied [29]. In addition, our goal is to find the similarity between tags, and these techniques are used in more complex tasks such as word analogy evaluation or concept categorization, among others [29].

Let us consider that V is the set of tags that form the vocabulary and s_{ij} the sentence containing a set of tags $\{t_l\} \subseteq V$ that the user $u_i \in U$ has given to the item $i_j \in I$. The word embedding techniques are applied to the tags in the vocabulary V to obtain a vector of tags v'_{t_o} as output. The following step is to transform these vectors into a one-dimensional space in order to have a single value for each tag. This is done by applying the PCA technique to each of the vectors. It consists of transforming the original variables into another set of variables, each of which is a linear combination of the former. The new variables are placed in order of highest to lowest variance. Thus, the most relevant ones are in the first places.

C. Incorporating tag embedding into the CF model

The result of processing the tags previously described will be used as complementary information to the ratings in order to characterize the users.

After performing tag embedding and applying PCA to the tag vectors, the tag embedding matrix $\mathbf{T}$ can be created. For the set of m users U and the set of n items I, the tag embedding matrix is defined as $\mathbf{T} := \overline{pca}_{i,j}$ where $\mathbf{T} \in M_{n\times m}(\mathbb{N})$ and $\overline{pca}_{i,j}$ is the average value obtained for each tag that user u_i assigns to item i_j by applying PCA to the output tag embedding vector. This matrix is used to calculate the similarity of users based on the tags they assign to the items. We denote this similarity based on tag embedding between the active user u_a and a user u_i as $sim_T(u_a, u_i)$. It can be computed using the cosine metric (eq. 5).

$$sim_T(u_a, u_i) = \frac{\sum_{j=1}^{n} \overline{pca}_{aj}\, \overline{pca}_{ij}}{\sqrt{\sum_{j=1}^{n} \overline{pca}_{aj}^2}\, \sqrt{\sum_{j=1}^{n} \overline{pca}_{ij}^2}} \quad (5)$$

The prediction of the rating that the active user u_a will assign to item i_j will be computed using the ratings-based similarities (eq. 1) and those based on tag embedding (eq. 5). $sim_T(u_a, u_i)$ will be used as a weight that modifies the ratings-based similarity in the equation used to make the predictions. Thus, equation 2 is transformed into the following:

$$pr_{aj} = \bar{r}_a + \frac{\sum_{i=1}^{k} sim_T(u_a,u_i)\, sim(u_a,u_i)(r_{ij} - \bar{r}_i)}{\sum_{i=1}^{k} |sim_T(u_a,u_i)\, sim(u_a,u_i)|} \quad (6)$$

Where both types of similarity are used to obtain the set of k neighbors $kNN = \{u_1, u_2, \ldots, u_k\} \subseteq U$ for the active user u_a. The predictions obtained with eq. 6 are used to generate the top-*N* recommendation lists in which the *N* items with the highest rating values for each user are included. The experimental study described in the following section validates the proposed method for both rating prediction and top-*N* recommendations.

IV. EXPERIMENTAL STUDY AND RESULTS

This study has been conducted to compare the proposal against some baseline methods. Since it is a neighborhood-based method, the main objective is to prove that the use of social tag embedding improves the classic CF approach using KNN (k Nearest Neighbors). Therefore, we have compared the proposed method (KNN Tag Embedding) with the user-based KNN CF approach. In addition, other CF methods based on matrix factorization have been tested, SVD and SVD++. This choice comes from the fact that they are two widely used methods that are indicated for both predicting ratings and recommending top-N lists, and can handle multivalued ratings, as our proposal, while other methods only handle binary values. In addition, as indicated at the end of section II, the tag-based methods in the literature have different objectives than those proposed in this paper or require additional information not available in most datasets, so we cannot include them in the comparative study.

The metrics used to evaluate rating prediction reliability are RMSE (Root-Mean-Square Error), MAE (Mean Absolute Error), NRMSE (Normalized RMSE), and NMAE (Normalized MAE). MAP (Mean Average Precision) and NDCG (Normalized Discounted Cumulative Gain) have been used for the evaluation of top-N recommendations. In all experiments, 5-fold cross validation was applied.

The datasets last.fm and movieLens have been used to validate the proposal. The first contains tagging and listening data from 1892 users of last.fm online music streaming platform. The number of played artists in the dataset is 17632 and the number of tags 11946. This dataset does not provide explicit ratings on the items, so they have been computed from the frequency of plays following the procedure described in [14]. The movieLens dataset contains data about 5-star rating and tagging activity from movieLens, a movie recommender system. It contains 3683 tags and ratings from 610 users on 9742 movies. The users in the dataset have rated at least 20 movies.

Since the objective of applying tag embedding in our work is to characterize users from the tags they assign to the items, it was applied considering that the window of words that constitute a sentence includes all the tags that a user has given to an item (an artist or a movie).

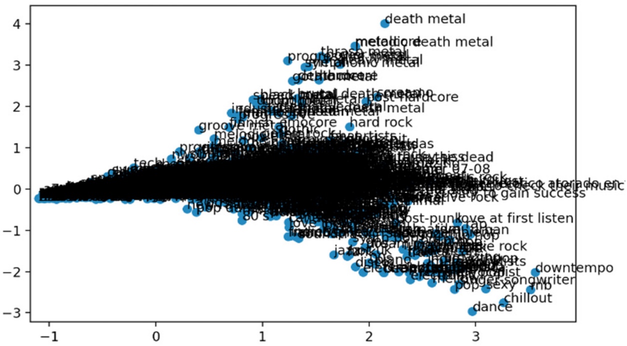

Figure 1. Tag embedding representation from the last.fm dataset

Word embedding was performed by using wor2Vec from the Gemsim library. The model was created setting the minimal number of word occurrence (*min_count* parameter) to 1 and the size of the output vector (*size* parameter) to 1500. *Min_count* was set to 1 to include as many tags as possible in order to better characterize users. The output vector size was the one that provided the best results. The package *sklearn.decomposition* from the *scikit-learn* library was used to apply PCA to the output vectors. Fig. 1 shows the representation in the two-dimensional space of the result of processing the tags contained in the last.fm dataset. A detail of an area of this representation focused on a subset of tags is showed in figure 2. To test the SVD and SVD++ methods, the Surprise library (http://surpriselib.com/) was used with the default configuration of the parameters, since changes in their values hardly modified the results.

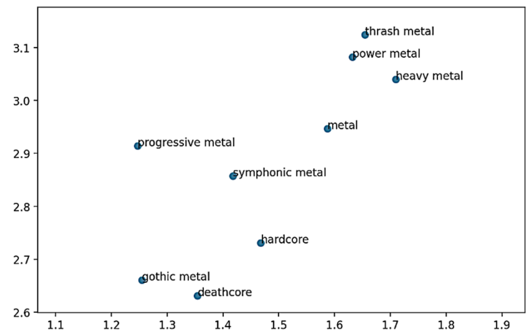

Figure 2. Detail of an area in figure 1 containing a subset of tags

Before comparing the results with all baseline methods, KNN-based methods were tested setting k to different values in order to determine the optimal number of nearest neighbors. The NRSME values obtained for the two datasets under study are shown in figure 3. While in the dataset last.fm the best results for KNN Tag Embedding are produced for k=5, in the dataset movieLens the values decrease as the number of neighbors increases, tending to stabilize from k=35. Therefore, those were the k values used for each dataset in the comparison with the rest of the methods.

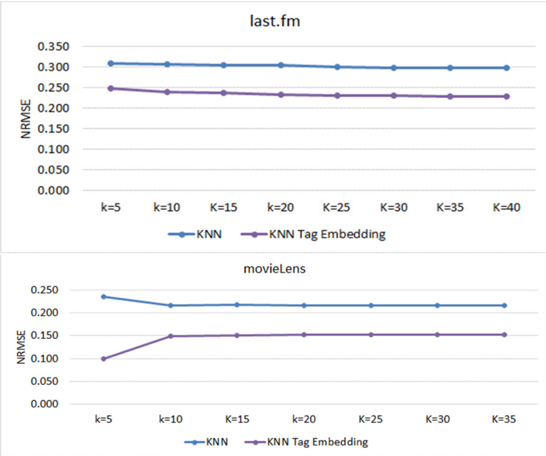

Figure 3. NRSME values obtained when applying KNN based methods on last.fm and movieLens datasets for different number of neighbors.

TABLE I. ERROR RATES OBTAINED WITH THE LAST.FM DATASET

Method	Dataset last.fm			
	RMSE	***NRMSE***	***MAE***	***NMAE***
KNN	1.195	0.298	0.945	0.236
SVD	1.043	0.261	0.922	0.231
SVD++	1.046	0.262	0.922	0.231
KNN Tag Embedding	**0.915**	**0.229**	**0.663**	**0.166**

TABLE II. ERROR RATES OBTAINED WITH THE MOVIELENS DATASET

Method	Dataset movieLens			
	RMSE	***NRMSE***	***MAE***	***NMAE***
KNN	0.939	0.235	0.779	0.195
SVD	0.650	0.163	0.505	0.126
SVD++	0.646	0.161	0.495	0.124
KNN Tag Embedding	**0.395**	**0.099**	**0.273**	**0.068**

First, the reliability of the model for rating prediction was evaluated. The error rates obtained with all methods tested with the last.fm and movieLens datasets are shown in tables I and II respectively. These results are represented graphically in Figure 4. When comparing the NRMSE values we can see that KNN Tag Embedding yields the lowest error rates, providing in the last.fm dataset an improvement of 12.56% over SVD++, 12.26% over SVD and 23.15% over KNN, while in the movieLens dataset the improvement was 38.51% over SVD++, 39,26% over SVD, and 57.87% over KNN.

In addition to proving that the proposed method performs better than baselines in predicting ratings, it is also necessary to show that it is valid for top-N recommendations, that is, for the lists of items with the highest ratings values, since those items are the ones that are recommended to the user. For this, MAP and NDCG rank-based metrics have been computed. The results obtained for N=10 are given in tables III and IV, and figure 5. The values of these metrics show that the best performance is achieved by the KNN Tag Embedding also for top-N recommendations. The percentages of improvement, as expected, are not as high for top-N recommendations as they are for rating predictions since MAP and NDCG metrics do not compare the values of the actual ratings with the predicted ones but are based on the ranking of the items in the list.

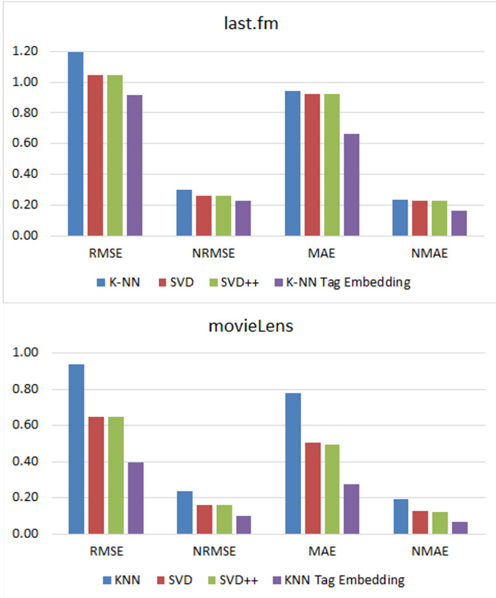

Figure 4. Error rates obtained with last.fm and movieLens.

TABLE III. MAP AND NDCG OBTAINED WITH THE LAST.FM DATASET

Method	Dataset last.fm	
	MAP	***NDCG***
KNN	**99.77%**	88.43%
SVD	91.67%	90.59%
SVD++	91.43%	89.81%
KNN Tag Embedding	**99.77%**	**91.64%**

TABLE IV. MAP AND NDCG OBTAINED WITH THE MOVIELENS DATASET

Method	Dataset movieLens	
	MAP	***NDCG***
KNN	69.33%	90.41%
SVD	64.89%	94.43%
SVD++	66.22%	94.76%
KNN Tag Embedding	**79.33%**	**94.92%**

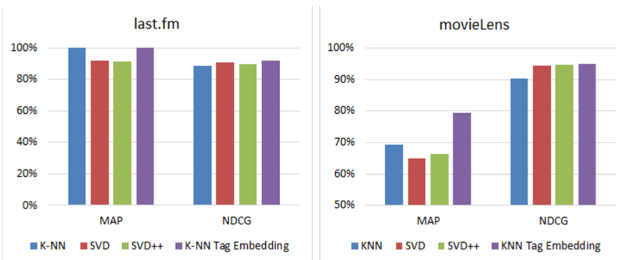

Figure 5. MAP and NDCG values obtained with last.fm and movieLens.

With the last.fm dataset, KNN and KNN Tag Embedding gave the same MAP value, although our proposal improved SVD and SVD++ results by 8.84% and 9.12% respectively. KNN Tag Embedding gave the highest value of NDCG with this dataset, outperforming KNN by 3.63%, SVD by 1.16%

and SVD++ by 2.04%. In the case of the movieLens dataset, the highest values of both MAP and NDCG were yielded by KNN tag Embedding. The increase in MAP was 14.42% over KNN, 22.25% over SVD and 19.8 over SVD++. Regarding NDCG, the increase was 4.99% over KNN, 0.52% over SVD and 0.17% over SVD++.

V. CONCLUSIONS

The intensive research being carried out in many areas on the exploitation of information from social networks, has been extended to the field of recommender systems. In this work, an approach focused on exploiting the information obtained from social tagging has been proposed. It is a CF scheme that extends the classical methods based on the nearest neighbors by including user similarities based on social tag embedding, a technique hardly used in the area of recommender systems. The proposal differs from others in the literature in the fact that it is a user-centered approach instead of being item-centered. In addition, tagging information is used to complement ratings unlike other methods that use tags to capture user preferences. It has been compared with other CF methods, one based on KNN and two based on matrix factorization. The results show that the proposed approach outperforms other methods in both rating prediction and top-N recommendations. A wider experimental study with a large number of baselines, including learning-to-rank methods, will be conducted as future work.

ACKNOWLEDGMENT

This research was funded by the Junta de Castilla y León, Spain, grant number: SA064G19.

REFERENCES

[1] J. B. Schafer, J. A. Konstant, and J. Riedl, "E-Commerce Recommendation Applications," Data Mining and Knowledge Discovery, 5, pp. 115-153, 2001.

[2] D. Sánchez-Moreno, A. B. Gil, M. D. Muñoz, V. F. López, and M. N. Moreno, "Recommendation of songs in music streaming services. Dealing with sparsity and gray sheep problems,", Advances in Intelligent Systems and Computing, vol. 619, pp. 206-213, 2017.

[3] D. Sánchez-Moreno, V.F. López, M.D. Muñoz, , A.L. Sánchez, M.N. Moreno, "Exploiting the user social context to address neighborhood bias in collaborative filtering music recommender systems," Information, 11(9), 439, 16 pages, 2020.

[4] O. Celma, Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space. Berlín Heidelberg, Germany: Springer, 2010.

[5] A. Mathes, "Folksonomies-Cooperative Classification and Communication Through Shared Metadata", 2004. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html.

[6] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A Neural Probabilistic Language Model," Journal of Machine Learning Research, 3 (2003) pp. 1137–1155, 2003.

[7] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Proc. of NIPS'13, vol. 2, December 2013, pp. 3111–3119.

[8] R. Jaschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme, "Tag recommendations in folksonomies," Proc. of PKDD'07, 2007, pp. 506-514.

[9] S. Sen, , J., Vig, and J. Riedl, "Tagommenders: Connecting Users to Items through Tags," Proc. WWW '09, pp. 671-680, 2009.

[10] J. H. Su, W. Y. Chang, and V.S. Tseng, "Personalized music recommendation by mining social media tags," Procedia Computer Science, 22 (2013), pp. 303-312, 2013.

[11] Z. Zhang, C. Liu, Y. Zhang, and T. Zhou, "Solving the cold-start problem in recommender systems with social tags," EPL 92 (2), 28002, 2010.

[12] D. Dolgikh, "Graph-based music recommendation approach using social network analysis and community detection method," Int. Conf. on Computer Systems and Technologies. ACM, pp. 221-227, 2015.

[13] F. Wang, L. Hu, R. Sun, J. Hu, and K. Zhao, "SRMCS: A semantic-aware recommendation framework for mobile crowd sensing," Information Sciences, 433-434 (2018), pp. 333-345, 2018.

[14] D. Sánchez-Moreno, M.N. Moreno-García, N. Sonboli, B. Mobasher, and R. Burke, "Inferring user expertise from social tagging in music recommender systems for streaming services", HAIS 2018, Lecture Notes in Artificial Intelligence, pp. 39-49, Springer, 2018.

[15] Y. Zuo, J. Zeng, M. Gong, and L. Jiao, "Tag-aware recommender systems based on deep neural networks," Neurocomputing, 204 (2016), pp. 51–60, 2016.

[16] Z. Xu, T. Lukasiewicz, C. Chen, Y. Mia, and X. Meng, "Tag-Aware Personalized Recommendation Using a Hybrid Deep Model," Proc. of IJCAI-17, pp. 3196-3202, 2017.

[17] M. de Gemmis, P. Lops, G. Semeraro, and P. Basile, "Integrating tags in a semantic content-based recommender," Proc. of RecSys'08, pp. 163-170, 2008.

[18] S. Siersdorfer, and S. Sizov, "Social Recommender Systems for Web 2.0 Folksonomies," Proc. of HT'09, pp. 261-270, 2009.

[19] N. Hariri, B. Mobasher, and R. Burke, "Context-aware music recommendation based on latent topic sequential patterns," Proceedings of the sixth ACM conference on Recommender systems, Dublin, Ireland, pp. 131-138, 2012.

[20] J. Gemmell, A. Shepitsen, B. Mobasher, and R. Burke, "Personalizing navigation in folksonomies using hierarchical tag clustering," Proc. of DaWaK '08, Berlin, Heidelberg: Springer-Verlag, pp. 196–205, 2008.

[21] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke, "Personalized recommendation in social tagging systems using hierarchical clustering," Proc. of RecSys '08, New York, NY, USA: ACM, pp. 259–266, 2008.

[22] H. Liang,Y. Xu,Y. Li, R. Nayak, and X. Tao, "Connecting users and items with weighted tags for personalized item recommendations," Proc. of HT 2010, NewYork, NY, USA: ACM, pp. 51–60, 2010.

[23] Z. Zhang, D. D. Zeng, A. Abbasi, J. Peng, and X. Zheng, "A random walk model for item recommendation in social tagging systems," ACM Trans. Manag. Inf. Syst., 4 (2), 8, 2013.

[24] H. Li, X. Diao, J. Cao, L. Zhang, and Q. Feng, "Tag-aware recommendation based on bayesian personalized ranking and feature mapping," Intell. Data Anal. 23 (3), pp. 641–659, 2019.

[25] D. Yang, L. Chen, J. Liang, Y. Xiao and W. Wang, "Social Tag Embedding for the Recommendation with Sparse User-Item Interactions," Proc. of ASONAM 2018, Barcelona, pp. 127-134, 2018.

[26] N. Liang, H. T. Zheng, J. Y. Chen, A. Sangaiah, and C. Z. Zhao, "TRSDL: tag-aware recommender system based on deep learning–intelligent computing systems," Appl. Sci., 8 (5), 799, 2018.

[27] R. Huang, N. Wang, C. Han, F. Yu, and L. Cui, "TNAM: A tag-aware neural attention model for Top-N recommendation," Neurocomputing 385 (2020), pp. 1–12, 2020.

[28] C. Dang, M.N. Moreno-García, F. De la Prieta, "Sentiment Analysis Based on Deep Learning: A Comparative Study," Electronics, 9 (3), 483, 29 pages, 2020.

[29] B. Wang, A. Wang, F. Chen, Y. Wang, C. Kuo, "Evaluating word embedding models: Methods and experimental results," APSIPA Transactions on Signal and Information Processing, 8, E19, 2019