2016 Joint 8th International Conference on Soft Computing and Intelligent Systems and 2016 17th International Symposium on Advanced Intelligent Systems

Collective Matrix Factorization using Tag Embedding for Effective Recommender System

Hanbyul Bang, Jee-Hyong Lee†
Department of Electrical and Computer Engineering
Sungkyunkwan University
Suwon, Republic of Korea
{hbyul91, john}@skku.edu

***Abstract*— Many people communicate each other through online community, SNS as Instagram, Facebook, etc. Most of these services annotate on their clips or pictures by using tags, which contain some information and can describe their contents. In this paper, we propose a new recommender system using word embedding with tag information and collective matrix factorization technique. By vectorizing tags that users annotated, we make user-tag matrix by merging tag vectors and factorize it together with user-item matrix. We show that this method effectively works through experiments.**

Keywords—recommender system; collaborative filtering; collective matrix factorization; word embedding; tag;

I. Introduction

As the age of web 2.0 begins, many people communicate each other on the internet and easily get informative data and share it without its owner. To access precise data which user want to get and to provide more informative data for user, most of web services annotate keywords on data, called "tag". It is a keyword assigned to clips, pictures or text in SNS, web browser like Facebook, Twitter etc. Contents with tags can describe in detail and give more information related to these contents more than contents without tags. Also, tags take on a role of the internet bookmark by connecting other web browsers.

Due to the fact that tags are recently working as important element on many services, a lot of researches have studied about tag recommender system and progressed it. Steffen Rendle et al [1]. proposed tag recommender system by Pairwise Interaction Tensor factorization technique which models the pairwise interactions between three elements; user, item and tag. Ralf Krestel et al [2]. recommended tag by using Latent Dirichlet Allocation (LDA) technique in order to figure out performance. In addition, not only tag recommender system but also recommender system for the other items by exploiting tag information have been studied. Huizhi Liang et al [3]. proposed a method that utilizes tagging information to personalized recommendations. Based on the distinctive three dimensional relationships among users, tags and items, they calculate new similarity score between users, tags and items. Bu Sung Kim et al [4]. improved performance of movie recommender system by Collective Matrix Factorization with user-movie matrix and user-tag matrix made of tag-movie relevance score. But these researches just utilized similarity or Bag-of-Words (BOW) as features, there are no consideration for exploiting semantic information about tags for recommender system. Also, for the reason that there are some hidden relations and latent information between three elements (user, item, tag) which can describe their characteristics effectively, we need to obtain additional latent information and utilize it.

In this paper, we propose a new recommendation model which uses collective matrix factorization technique and utilizes word embedding. This model adopts word embedding for tags what users attach to items, and from now on, we call this "Tag Embedding". We will predict the unknown ratings in user-item matrix by collective matrix factorization with user-tag matrix that is constructed by Tag Embedding.

The rest of this paper is organized as follows. Section 2 provides a brief explanation about background knowledge of our proposed method like Collaborative filtering recommender system, collective matrix factorization and word embedding. In Section 3, we start to explain how proposed method works and how imply some techniques to our recommender systems. Section 4 is for experiments. In here, we explain what dataset is used, how experiment designs are setting. And we report the experimental results at the end of Section 4. Finally Section 5 gives a conclusion about our work.

II. Background

A. Collaborative Filtering Recommender System

Recommender system is one of information filtering system that predicts users' preferences about items in order to predict items a target user will prefers [5]. This can predict user's preferences by analyzing data like users' ratings, tendency, bias about item, etc. The most prominent approach to generate recommendation is Collaborative Filtering (CF) method. CF recommends items which is expected target user will prefer, based on idea that user who had similar tastes in the past, will have similar tastes in the future [6]. One of the CF models is nearest-neighbor model. It can be either user-based or item-based. They calculate similarity score like Pearson correlation coefficient, cosine similarity between users or items to predict unknown rating. Matrix Factorization models are efficient in high sparsity data with a low rank approximation by minimizing a loss function. Because

978-1-5090-2678-4/16 $31.00 © 2016 IEEE
DOI 10.1109/SCIS&ISIS.2016.69

of their high performances, they are used more than nearest-neighbor model. But general matrix factorization model just only uses ratings or preference of users for items to make user-item matrix, without using the other useful information to improve recommendation performance.

B. Collective Matrix Factorization

Through changes of web, web 2.0 allows user to get informative data easily and to interact each other with a large amount of associated data. For the above reason, it is almost impossible that such data are represented by a single adjacency matrix [7]. The Collective Matrix Factorization (CMF) is one of prominent decomposition techniques, which factorize several matrices together to improve performance. CMF simultaneously factorize matrices with sharing some sub-matrices if they have meaningful relations [8]. Furthermore, it has been proven that recommender system using CMF has higher performances and can reduce sparsity of user-item matrix [9].

C. Word Embedding

In Natural Language Processing (NLP), most tasks represent words by using one-hot representation. This method represents word as vector filed with all “0”s, except for only one “1” at the position related with that word, and the dimension of the vector is vocabulary size. For example, if you have 5 words, you can represent 4th word as vector “$v =< 0, 0, 0, 1, 0 >$” by one-hot representation. However, vectors of words represented by one-hot encoding method do not contain any similarity or dis-similarity information between words, it is impossible to represent relations between words. Moreover, if vocabulary size is large, vector becomes high-dimensional and it has sparsity problem.

To solve these problems, Ronan Collobert et al [10]. And Tomas Mikolov et al [11]. represent words as dense vector by using neural-network language modeling. These representation methods are called “word embedding”. Many researches related to word embedding show that this distributed vector representation (word embedding) significantly outperform past NLP researches [12, 13]. Through word embedding, words are mapped to vectors that consist of real numbers. It can represents the latent semantic and syntactic meaning of words.

One-hot Encoding		Word Embedding
1, 0, 0, 0, 0, 0, 0	the	0.22, 0.67, 0.00, 0.05, 0.21, 0.12
0, 1, 0, 0, 0, 0, 0	a	0.20, 0.64, 0.05, 0.03, 0.21, 0.12
0, 0, 1, 0, 0, 0, 0	cat	0.95, 0.87, 0.12, 0.45, 0.54, 0.77
0, 0, 0, 1, 0, 0, 0	on	0.13, 0.07, 0.20, 0.65, 0.11, 0.67
0, 0, 0, 0, 1, 0, 0	dog	0.95, 0.88, 0.10, 0.40, 0.59, 0.77

Fig. 1. Word representation

Fig. 1. shows that 2 kinds of word representation, one-hot encoding and word embedding. Vectors by one-hot encoding consist of zeros or one, vectors by word embedding consist of real numbers. Although words “cat” and “dog” are similar in terms of animals, you can see that one-hot encoding cannot shows the similarity between two words in their representation but word embedding can do. So, in this paper, we represent tag by using word embedding, thereby we extract latent semantic information and utilize it.

III. Proposed Method

In this section, we proposed recommendation system which uses collective matrix factorization technique with tag embedding. Proposed method takes 3 steps. Firstly, we collect tag dataset and classifies these tag data as users who tagging it. Next, this method converts tags collected at previous step into vectors through word embedding and constructs user-tag matrix using these vectors. Finally, we implement collective matrix factorization between user-item matrix and user-tag matrix. To give a detailed explanation of CMF process, proposed method factorizes user-item matrix and user-tag matrix separately into two sub-matrices with sharing one sub-matrix. After factorizing these matrices, we define an error function and implement learning to minimize error. And then, we can make a result of predicted ratings by multiplying two sub-matrices.

A. Problem setup and Notation

The goal of our proposed method is to effectively recommend items to user by collective matrix factorization, and one of factorized matrices is constructed in the process of tag embedding.

To help understanding our proposed method, firstly we define the representation of matrix, vector and set as below:

$$U = \{u_1, u_2, \cdots, u_m\} \tag{1}$$

$$I = \{i_i, i_2, \cdots, i_n\} \tag{2}$$

$$\boldsymbol{R} = \{r_{u,i}\}, u \in U,\ i \in I, (m \times n) \tag{3}$$

$$Tvec_u =< t_u^1, t_u^2, \cdots, t_u^d > \tag{4}$$

$$\boldsymbol{T} = \begin{matrix} Tvec_1 \\ \vdots \\ Tvec_m \end{matrix}, \ (m \times d) \tag{5}$$

U is a set of m users, I is a set of n items, $\boldsymbol{R}$ wrote by bold means user-item matrix, size $m \times n$, $Tvec_u$ is vector of tags for user u with d dimension. Also user-tag matrix can be represented by a set of vector $Tvec_u$.

B. User-Tag Matrix

We utilize latent semantic information of tag to improve recommendation performance. Through Tags which are annotated on item, we can guess characteristics of that item. And also we can guess style of users who tagging on items, because this reflects users' style, preference, and so on.

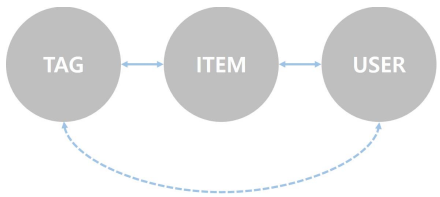

Fig. 2. Relations of user, item and tag

The procedure of constructing user-tag matrix is as fig. 3.

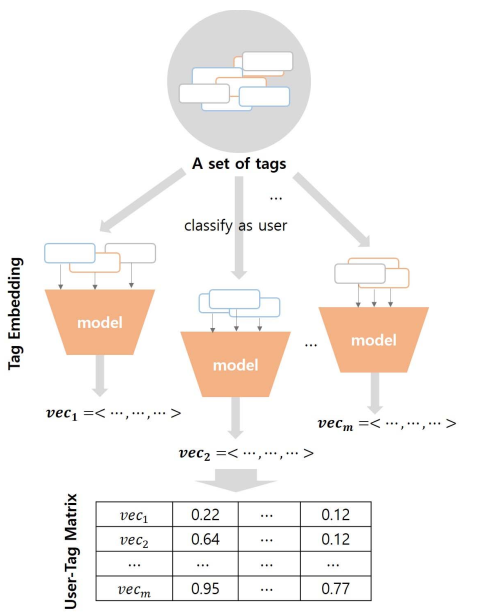

Fig. 3. Flow of constructing user-tag matrix by word embedding

To construct user-tag matrix by word embedding, firstly we classify a set of tags as user who annotating that. And next, we do tag embedding to each user through some embedding model. In here, we exploit average values to represent users' tag vector. Eq. 6. represents users' tag vector:

$$Tvec_u = \frac{1}{N_u}\sum_{n=1}^{N_u} tag_{u,n} \tag{6}$$

N_u is the number of words user u tagging, and $tag_{u,n}$ is a vector of n^{th} tag which user u writes. As summing $tag_{u,n}$ of user u and dividing that summation by the number of words, we can calculate average and get d- dimensional user u's tag vector. And next we engraft all $Tvec_u$ of all users into a $m \times d$ user-tag matrix.

C. Collective Matrix Factorization

The Collective Matrix Factorization (CMF) technique extracts meaningful information from associated data by factorizing each matrices into sub-matrices with sharing the same sub-matrix [14]. In this paper, we use two matrices, which are user-item matrix and user-tag matrix constructed by tag embedding. Firstly, we factorize user-item matrix into a product of user-specific latent factor sub-matrix X and item-specific latent factor sub-matrix Y^T, and factorize user-tag matrix into a product of and X and Z^T which is tag's latent semantic sub-matrix. The Structure of factorized matrices in proposed method as follows:

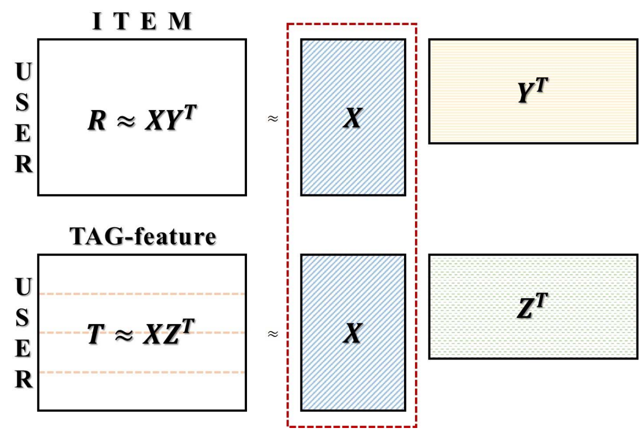

Fig. 4. Structure of factorized matrices

D. Learning Error function

To minimize error between predicted values and real values, we formulate our error function as Eq. 7.

$$E(X, Y, Z) = \frac{1}{2}\|I^\circ(R - XY^T)\|_F^2 + \frac{\alpha}{2}\|T - XZ^T\|_F^2 + \frac{\beta}{2}(\|X\|_F^2 + \|Y\|_F^2 + \|Z\|_F^2) \tag{7}$$

$\|\cdot\|_F^2$ denotes the Frobenius norm, and I is indicator matrix. Last term in error function helps regularizing and avoiding overfitting to the training data.

We can represent our objective as follow Eq. 8., and get predicted values to solve this:

$$\underset{X,Y,Z}{\operatorname{argmin}}\, E(X, Y, Z) \tag{8}$$

The problem with Eq. 8. can be solved using Gradient Descent Method (GDM). The below equations are gradients for each variable:

$$\begin{aligned} \nabla_X E &= [I^\circ(XY^T - R)]Y + \alpha(XZ^T - T)Z + \beta X, \\ \nabla_Y E &= [I^\circ(XY^T - R)]^T X + \beta Y, \\ \nabla_Z E &= \alpha(XZ^T - T)^T X + \beta Z \end{aligned} \tag{9}$$

We use Eq.s 9. for gradient descent method. The details of the GDM are given in fig. 5.

Gradient Descent Method for CMF
inputs:
$\boldsymbol{R}$: **user-item matrix**
$\boldsymbol{T}$: **user-tag matrix**
$\boldsymbol{I}$: **Indicator matrix**
outputs: $\boldsymbol{XY^T}$

1. **Initialize X, Y, Z randomly**
2. $\boldsymbol{t = 1}$;
3. $\boldsymbol{While(t < max - epoch}$ && $\mathrm{E}_t - \mathrm{E}_{t+1} > \epsilon) \boldsymbol{do}$
4. **Get the gradients by equation (9).**
5. $\boldsymbol{\gamma = 1}$;
6. $\boldsymbol{While}(\mathrm{E}(X_t - \gamma\nabla_{X_t}, Y_t - \gamma\nabla_{Y_t}, Z_t - \gamma\nabla_{Z_t}) > \mathrm{E}(X_t, Y_t, Z_t))\ \boldsymbol{do}$
7. $\boldsymbol{\gamma = \frac{\gamma}{2}}$;
8. $\boldsymbol{X_{t+1} = X_t - \gamma\nabla_{X_t}}$;
9. $\boldsymbol{Y_{t+1} = Y_t - \gamma\nabla_{Y_t}}$;
10. $\boldsymbol{Z_{t+1} = Z_t - \gamma\nabla_{Z_t}}$;
11. $\boldsymbol{t = t + 1}$;
12. **return** $\mathbf{XY^T}$

Fig. 5. Gradient Descent Method Algorithm

IV. EXPERIMENTS

A. Data Set

To demonstrate performance of our proposed method, we use MovieLens dataset that includes not only user, movie data but also tag data. We only use users' data whose rating about movie and tag exist both. One tag data is matched only one rating of one movie by one user. The number of dataset is in table (1).

TABLE I. THE NUMBER OF DATASET

	User	Movie	Tag
Number	**2,320**	**10,190**	**10,606**

B. Experiment Design

To modeling tag vectors, we select Word2vec model that consists of shallow, two-layer neural networks. This model relies on skip-grams or continuous bag of words (CBOW) to create word embedding [15]. In order to measure performance of our proposed method, we set 80% of users' rating data about movie as training data and 20% as test data. We implement 3 experiments as follows:

1) Sigular Value Decomposition

First is applying one of popular decomposition technique, Singular Value Decomposition (SVD). We implement this as a baseline.

2) CMF-BOW

Next is CMF with user-tag matrix constructed by bag-of words (BOW). We call this method as CMF-BOW. We use CMF-BOW as second baseline.

3) Proposed method

Final experiment is our proposed method, CMF with tag-embedding.

C. Evaluation Metrics

To evaluate performances of our proposed method, we use Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as a performance evaluation metric.

$$RMSE = \sqrt{\frac{1}{n}\sum_{(u,i)\in S}\left(r_{u,i} - \hat{r}_{u,i}\right)^2} \tag{10}$$

$$MAE = \frac{1}{n}\sum_{(u,i)\in S} | r_{u,i} - \hat{r}_{u,i} | \tag{11}$$

S represents a set of test data and $\hat{r}_{u,i}$ is predicted rating value. Both evaluation metric have better performance when having small values.

D. Experiment Results

Table (2) shows results of our 3 experiments.

TABLE II. COMPARISIONS ON EACH METHODS

	RMSE	MAE
SVD	0.8378	0.6426
CMF-BOW	0.8168	0.6242
Proposed method	**0.7946**	**0.6047**

As shown in above results, difference between SVD method and proposed method is more than 0.04 on RMSE and more than 0.03 on MAE. Moreover, proposed method outperforms CMF-BOW method. In comparison between first experiment (SVD) and CMF-BOW, proposed method (second and third experiment) which utilize tag information, we can find that tag information helps improving recommendation performances. In comparison between CMF-BOW and proposed method, we can see recommendation with semantic tag information outperforms than without semantic tag information.

V. CONCLUSION

In this paper, we proposed the collective matrix factorization method with tag embedding for recommender system. We shows that our proposed method is outperformed in experiments and useful to who uses tag services. Our experimental results tell us two facts. First is, user, item and tag have some relations each other, so they can improve their performance by collective matrix factorization technique with sharing additional information each other. Second fact is, our proposed method exploits latent semantic information of tag by word embedding, so it outperforms the other method using tag without semantic information. In the future, we will study word embedding more and suggest applicable word embedding-based recommender system.

ACKNOWLEDGMENT

This research was supported by ICT R&D program of MSIP/IITP (10041244, Smart TV 2.0 Software Platform) and was supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2016

REFERENCES

[1] S. Rendle, and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," *Proceedings of the third ACM international conference on Web search and data mining*. ACM, pp. 81-90, 2010.

[2] R. Krestel, P. Frankhauser, and W.Nejdl, "Latent Dirichlet Allocation for Tag Recommendation," *Proceedings of the third ACM conference on Recommender systems*. ACM, pp. 61-68, 2009.

[3] Goldberg, Yoav, Levy, and Omer. "word2vec Explained: Deriving Mikolov et al.' s Negative-Sampling Word-Embedding Method," arXiv preprint, 2014.

[4] H. Liang, Y. Xu, Y. Li, and R. Nayak, "Collaborative Filtering Recommender Systems Using Tag Information," *Web Intelligence and Intelligent Agent Technology,* Vol. 3, pp. 59-62, *2008.*

[5] BS Kim, H Kim, J Lee, and JH Lee, "Improving a Recommender System by Collective Matrix Factorization with Tag Information," *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*. IEEE, pp. 980-984, 2014.

[6] R. Francesco, R. Lior, and S, Bracha, "Introduction to Recommender Systems Handbook, Recommender Systems Handbook," Springer, pp. 1-35, 2011.

[7] S. Badrul, K. George, K. Joseph, and R. John, "Item-based Collaborative Filtering Recommendation Algorithms," Proceedings of the 10th international conference on World Wide Web. ACM, pp. 285-295, 2001.

[8] Guillaume Bouchard, Shengbo Guo, and Dawei Yin, "Convex Collective Matrix Factorization," Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, vol. 31, pp.144-152, 2013.

[9] Ajit P. Singh and Geoffrey J. Gordon, "Relational learning via collective matrix factorization," Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 650-658, 2008.

[10] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang, "Transfer Learning in Collaborative Filtering for Sparsity Reduction," AAAI, vol.10, pp. 230-235, 2010.

[11] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," In ICML, 2008.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean, "Distributed representations of words and phrases and their compositionality," In NIPS, 2013.

[13] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," Journal of Machine Learning Research, vol.3, pp. 1137-1155, 2003.

[14] H. Schwenk, "Continuous space language models," Computer Speech and Language, vol. 21, 2007.

[15] Hanbyul Bang, HyeWoo Lee, and Jee-Hyong Lee, "The YouTube Video Recommendation by Considering Subscription Information," ISIS 2015 The 16th International Symposium on Advanced Intelligent Systems, pp. 825-833, 2015.