UAFA: Unsupervised Attribute-Friendship Attention Framework for User Representation

Yuchen Zhou[1,2], Yanmin Shang[2(✉)], Yaman Cao[2], Yanbing Liu[2], and Jianlong Tan[2]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

{zhouyuchen,shangyanmin,caoyanan,liuyanbing,tanjianlong}@iie.ac.cn

Abstract. The problem of user representation has received considerable attention in recent years. A variety of social networks include not only network structures (friendships) but also information about users' attributes. Previous studies have explored the integration of the two information to encode users. However, these methods focus on how to fuse the target user's friendships as a whole with its attribute information to get its representation vector, without considering the inside information of friendships, that is the influence of intimacy difference between the target user and its each friend on its representation vector. In addition, most of the above methods are supervised, which can only be applied to limited social networks analysis tasks. In this paper, we investigate a novel unsupervised method for learning the user representation by considering the influence of intimacy difference. The proposed methods take both the users' attributes and their friendships into consideration with attribute-friendship attention network. Experimental results demonstrate that the user vectors generated by the proposed methods significantly outperform state-of-the-art user representation methods on two different scale real-world networks.

Keywords: Social network analysis · User representation · Attention mechanism · User embedding

1 Introduction

Nowadays, various applications of social networks have penetrated into all aspects of life and work, and the research based on social networks is also in full swing. How to learn user representation is a key issue in social network research. Inspired by the idea of network embedding, user embedding is one of the most effective ways to learn user representation in recent years. User embedding not only can obtain distributed user representation vectors, but also solve the dimension disaster problem of user representation.

C. Douligeris et al. (Eds.): KSEM 2019, LNAI 11775, pp. 155–167, 2019.
https://doi.org/10.1007/978-3-030-29551-6_14

At present, the methods of learning user representation vectors by user embedding can be roughly divided into two categories. The first category uses user's single data to obtain user's representation vector, such as user attributes, user friendship, user published information, etc. Although this kind of approaches can acquire user's distribution representation vector, it only considers the user's single information and ignores the improvement of the accuracy of the user's multiple information to its representation vectors. The second category uses user's mixed data, that is, the above two or more types of data are used simultaneously to achieve user's representation vector. This article focuses on the mix of user attributes and friendship. At present, many studies have used this kind of mixed information to obtain user's representation vectors. Compared with the first kind of methods, the performance advantages of these methods have also been proved in practice. However, there are still some shortcomings in these methods: (1) the idea of these methods is to fuse multiple friend relationships of target user as a whole with its attribute information, ignoring the influence of the differences between friend relationships on the target user's representation vectors. (2) Influenced by whether the data is labeled or not, most these approaches are supervised, which greatly limits the application of these methods in different social network analysis tasks.

In order to solve above problems, this paper proposes an unsupervised user representation method (UAFA) which integrates user's attributes and user's friendships, and this method use attention mechanism to model the influence of user relationship differences on target user representation vector. Specifically, each user has two representation vectors, which are derived from user attributes and user relationship information respectively. We assume that each user's two representation vectors are similar. Based on above assumption, firstly, we design an encoder to encode each user into vector by using its own attributes, which is called ego-representation vector. Secondly, we adopt attention mechanism to transform user's friendship into vector which is called friendship-representation vector. Finally, we align the friendship-representation vector and ego-representation vector for each user to make the two vectors as similar as possible. In our method, we explored three concrete models which respectively adopt Convolutional Neural Network, Recurrent Neural Network and Deep Neural Network as encoders.

To summarize, we make the following contributions:

- As far as we know, this paper first focuses on the influence of user's friendship differences on its presentation vectors, and proposes an attribute-friendship attention framework to model the above influence. In addition, the framework is unsupervised which can eliminate the impact of data labels on model adaptability.

- Based on UAFA, we proposed three concrete models and empirically evaluate them for three tasks (gender prediction, occupation prediction and friend recommendation) on several real-world datasets.

2 Related Work

User Embedding is a special case of network embedding. Existing methods for user embedding representation learning include single data approaches and mix data approaches.

2.1 Single Data Approaches

UE model [3] and MUVR model [5] are typical single data approaches, they regard users' relations as the context of user and adopt CBOW or Skip-Gram model to learn users' representation. NMCF model [4] uses RNN to encode the content that users' post and adopts mean pool to reduce dimensions to learn users' representation. Although those approaches can acquire user representation, they only use single type of data by ignoring richer information.

2.2 Mix Data Approaches

SWE model [11], User2Vec model [10] and SBFTE model [7] integrate users' relations and content they post to acquire users' representation. While SocialEM model [9] and JUERL model [8] use users' attributes and post to acquire users' representation. All the above models are supervised. LME model [2] adopts Generalized Canonical Correlation Analysis to combine users' relations with post to generate users' representation. Although LME model is an unsupervised way, it ignores the internal differences in users' relationships. In this paper, we will propose an unsupervised user representation method by considering mix data at the same time.

3 UAFA

As we all know, on the one hand, users' attribute information on social networks reflects their essential representation. On the other hand, according to the homophily theory, users' representation are influenced by their friends' representations. In this paper, we consider users' representations from the above both perspectives, and propose an unsupervised attribute-friendship attention framework (UAFA). This framework includes two components, ego-representation and friendship-representation.

3.1 Ego-Representation

As we know, different attributes have different data formats, for example, "sex" is usually filled by one word (male or female) while "university" is always filled by two or more words. For each attribute, in order to obtain attribute vector with the same dimension, we adopt pre-trained word vectors. Specific, for attributes that have only a single word, we directly use its word vector as the attribute vector. For those attributes with more than one word, we use the weighted average vector of these words as the attribute vector. In this way, each attribute can be transformed into a d-dimensional vector with Look-up table.

Suppose user u_k has m attributes, according to above process, we can get m attribute vectors of u_k, denoted as $a_{k,1}, a_{k,2}, \cdots, a_{k,m}$. Here, $a_{k,t}$ represents the vector of the t-th($1 \leq t \leq m$) attribute and its dimension is d. For convenience, we use $A_k \in R^{m \times d}$ to represent u_k attributes vector matrix, and $a_{k,t}$ is a row element of A_k. We input each row element of the matrix A_k into LSTM one by one, and after encoding, we use hidden layer encoding as user u_k ego-representation ur_k (the dimension is D). It can be expressed as follows:

$$f_{k,t} = sigmoid(W_f \cdot [h_{k,t-1}, a_{k,t}] + b_f), t \in [1, m] \tag{1}$$

$$i_{k,t} = sigmoid(W_i \cdot [h_{k,t-1}, a_{k,t}] + b_i), t \in [1, m] \tag{2}$$

$$\tilde{f}_{k,t} = tanh(W_C \cdot [h_{k,t-1}, a_{k,t}] + b_C), t \in [1, m] \tag{3}$$

$$C_{k,t} = f_{k,t} * C_{k,t-1} + i_{k,t} * \tilde{f}_{k,t}, t \in [1, m] \tag{4}$$

$$o_{k,t} = sigmoid(W_o \cdot [h_{k,t-1}, a_{k,t}] + b_o), t \in [1, m] \tag{5}$$

$$h_{k,t} = o_{k,t} * tanh(C_{k,t}), t \in [1, m] \tag{6}$$

$$ur_k = h_{k,m} \tag{7}$$

The structure of ego-representation component is shown in Fig. 1.

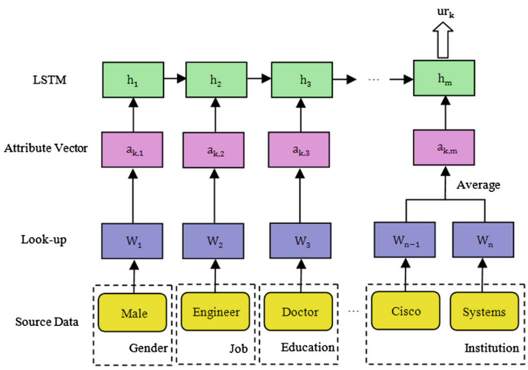

Fig. 1. Structure of Ego-representation component

3.2 Friendship-Representation

In this part, we consider the influence of user's friend representation on its representation, so as to obtain another user representation vector, namely friendship-representation. Suppose u_k has N friends denoted as $\{u_{k,1}, u_{k,2} \cdots, u_{k,N}\}$, according to the calculation process in the preceding section, we can obtain each friend's ego-representation vector $ur_{k,1}, ur_{k,2}, \cdots, ur_{k,N}$. For convenience, we use $UF_k \in R^{N \times D}$ to represent u_k's friend representations vector matrix, and $ur_{k,j(1 \leq j \leq N)}$ is a row element of UF_k. UF_k is the input of friendship-representation component.

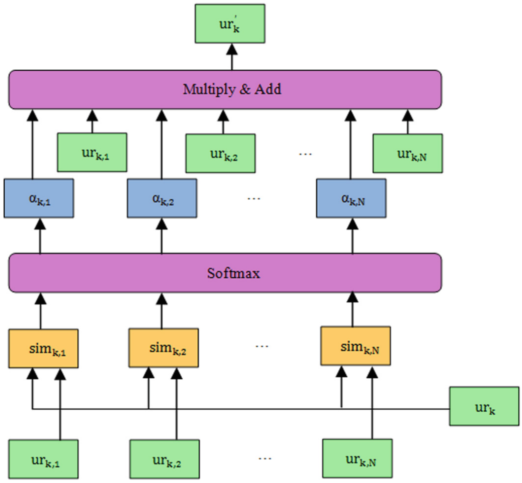

Fig. 2. Structure of friend-representation component

According to the homophily theory, the user's representation is influenced by his friends' representations. How to accurately model this influence is the problem we focus on. The most intuitive approach is to consider that each friend has the same impact on the user, that is, to use the average vector of all friend representations as the user's representation. However, this approach ignores the intimacy difference between user and different friends. In this paper, we model the influence by using attention mechanism [1]. We denote the friendship-representation of user u_k as ur'_k, it can be obtained as follows:

$$ur'_k = Attention(u_k; u_{k,1}, \cdots, u_{k,N}) = \sum_{j=1}^{N} \alpha_{k,j} * ur_{k,j} \tag{8}$$

Where $\alpha_{k,j}$ is weight coefficient, and $\alpha_{k,j}$ represents the impact of $u_{k,j}$ on u_k, it can be expressed as (9).

$$\alpha_{k,j} = Softmax(Sim_{k,j}) = \frac{e^{Sim_{k,j}}}{\sum_{i=1}^{N} e^{Sim_{k,i}}} \tag{9}$$

We adopt multiplicative attention and the similarity function $Sim_{k,j}$ can be expressed as (10).

$$Sim_{k,j} = Similarity(u_k, u_{k,j}) = ur_k^T W_a ur_{k,j} \tag{10}$$

Where $W_a \in R^{D \times D}$ is the parameter of attention model. The structure of friendship-representation component is shown in Fig. 2.

3.3 Objective Function

Based on the assumption that each user's ego-representation and friendship-representation should be as similar as possible, we use cosine distance to measure similarity and define the similarity between ur_k and ur'_k as the loss function J. More formally, the objective of UAFA is to minimize the loss function J. More formally, the objective of UAFA is to minimize the loss function J, U is the total number of users.

$$MinJ = Min\sum\nolimits_{k=1}^{U} sim(ur_k, ur'_k) = Min\sum\nolimits_{k=1}^{U} cosine(ur_k, ur'_k) \tag{11}$$

4 Experiments Design

We evaluate our method on two real-world datasets (Table 1):

- Google+: We select 1256 users with full attributes and construct a new closed social circle, and this social circle contains 54857 relations.
- Sina Weibo: We sample 16772 users with full attributes with their attributes and collect their friendships (following and followed).

Table 1. Description of two data sources

Data source	User	Relations	Average friends of each user
Google+	1256	54857	127
Sina Weibo	16772	1324988	79

To transform users' attributes into vectors, we use a set of trained English word vectors generated by using Word2Vec model on 12.7 GB Wikipedia data, and a set of trained Chinese word vectors generated by using Word2Vec model on 4.1 GB Baidu Encyclopedia.

4.1 Baselines

Word2Vec. This is the current mainstream approach to learn word embedding, it contains CBOW model and SkipGram model. In recent years, Word2Vec is also introduced in social network for user representation.

Support Vector Machine. As a classic classifier, in experiment, we use it to predict the gender and occupation attribute.

Random Forest. As a classic decision tree classifier, in the experiment, we use it to predict the gender and occupation attribute.

UE Model. This is a user embedding method proposed by Chen in 2016 [3], and the parameters of UE used in our paper are the same as [3].

DeepWalk. It is the first paper which introduces word embedding into network embedding [6], and generates node sequence by random walk. In experiments, we set the random walk length is 80 and the window size is 10.

Node2Vec. Different from DeepWalk, Node2Vec makes the node sequence contain local and macro information simultaneously by introducing two parameters p and q to achieve a balance in BFS and DFS. In experiments, we set the p is 0.5 and q is 2.

DNN-ATT (DNN). This is the first kind of concrete model of the UAFA, DNN-ATT. This model uses DNN as encoder to generate user ego-representation and uses attention mechanism to model the difference in the impact of friends on users. Moreover, we design a comparative model to verify the effectiveness of attention mechanism, DNN-AVE, which uses average strategy to obtain the friendship-representation. To highlight the use of attention mechanisms in model, we record DNN-AVE as DNN. In the above two models, the DNN network layout is 4, the middle two layers size is 512, and the last layer size is 256.

CNN-ATT (CNN). This is the second kind of concrete model of the UAFA, CNN-ATT. This model uses CNN as encoder to generate user ego-representation and uses attention mechanism to model the difference in the impact of friends on users. Similarly, we design a comparative model with CNN and average strategy, abbreviated as CNN. In the above two models, we define two convolution layers, use ReLU as activation function, adopt the max pool to reduce dimension, and use a linear function to normalize output vectors into 256 dimensions.

LSTM-ATT (LSTM). This is the third kind of concrete model of the UAFA, LSTM-ATT. This model uses LSTM as encoder to generate user ego-representation and adopts attention mechanism to model the difference in the impact of friends on users. Similarly, we design a comparative model with LSTM and average strategy, abbreviated as LSTM. In the above two models, the hidden layer size is 256.

4.2 Tasks Setup

In this paper, we select three tasks to demonstrate the effectiveness of UAFA: user gender prediction, user occupation prediction and friend recommendation.

Gender Prediction. The user's gender prediction task is to infer the missing gender attribute of a user. We partition the dataset into 10 folds, 1 fold for testing and other 9 folds for training. For each user in test set, we select the closest user in the training set and take its gender as the gender of the user in the test set. In this task, we adopt precision, recall and F1 score as the evaluation criteria.

Occupation Prediction. The user's occupation prediction task is to infer the missing occupation attribute of a user. Similar to gender prediction task, we also partition the dataset into 10 folds, 1 fold for testing and other 9 folds for training. According to these datasets, we divide occupations into 8 categories in advance. For each user in test set, we select the closest user in the training set and take its occupation as the occupation of the user in the test set. In this task, we adopt precision, recall and F1 score as the evaluation criteria.

Friend Recommendation. Friend recommendation aims to find other users that one user tend to make friends with. We partition the dataset into 10 folds, 1 fold for testing and other 9 folds for training. For each user in test set, we select the top K users closest to it as candidate friend.

5 Performance Comparison

For the above three tasks, we design the comparison methods from the perspective of using single data and mixed data. In specific, CBOW, Skip-Gram, DeepWalk, Node2Vec and UE model only use users' friendship, Random Forest and SVM only use users' attributes. In general, all of the above methods use a single type of data to represent users. In contrast, LSTM-ATT (LSTM), CNN-ATT (CNN), DNN-ATT (DNN) are the approaches which use mix type of data, and combine users' attributes and friendships to represent users.

5.1 Gender Predictions

Superiority of Mixed Data. As can be seen from Table 2, SVM is the best method to use single data and LSTM is the best method to use mixed data. For single data, the methods using attribute information (SVM and RF) are more suitable for gender prediction than methods with friendship information (CBOW, Skip-Gram, DeepWalk, Node2Vec and UE model), then, whether the combination of attribute information and friendship information will improve the performance of gender prediction? The performance of LSTM is obviously a good answer to this question. The performance of LSTM in gender prediction task is significantly higher than that of the former. This reveals that more comprehensive and accurate user representations can be achieved by using both user attributes and friend relationships. However, Table 2 also shows that the prediction precision of CNN and DNN is not higher than that of single data methods. The reason is that these two methods do not capture the long-term dependencies of attributes and relationships as LSTM does. Nevertheless, the performance of CNN and DNN is not much lower than the methods which only use single data.

Table 2. Performance comparison between single and mixed data models

Model	Google+			Sina Weibo		
	Precision	Recall	F1	Precision	Recall	F1
CBOW	0.642	0.613	0.627	0.808	0.783	0.795
DeepWalk	0.683	0.664	0.673	0.821	0.801	0.811
Node2Vec	0.701	0.682	0.691	0.857	0.834	0.845
Skip-Gram	0.734	0.709	0.721	0.843	0.826	0.834
UE Model	0.744	0.711	0.727	0.865	0.841	0.853
Random Forest	0.755	0.732	0.743	0.892	0.873	0.882
SVM	**0.764**	**0.748**	**0.756**	**0.906**	**0.886**	**0.896**
CNN	0.604	0.587	0.595	0.714	0.693	0.703
DNN	0.714	0.693	0.703	0.865	0.841	0.853
LSTM	**0.853**	**0.837**	**0.845**	**0.956**	**0.927**	**0.941**

Table 3. Performance comparison among methods with attention mechanism

Model	Google+			Sina Weibo		
	Precision	Recall	F1	Precision	Recall	F1
CNN	0.604	0.587	0.595	0.714	0.693	0.703
CNN-ATT	**0.623**	**0.605**	**0.614**	**0.756**	**0.733**	**0.744**
DNN	0.714	0.693	0.703	0.865	0.841	0.853
DNN-ATT	**0.726**	**0.707**	**0.716**	**0.879**	**0.852**	**0.865**
LSTM	0.853	0.837	0.845	0.956	0.927	0.941
LSTM-ATT	**0.875**	**0.858**	**0.866**	**0.977**	**0.953**	**0.965**

Effect of Attention Mechanisms. Table 3 shows the performance of models with hybrid data using attention mechanism. Compared with the methods without attention mechanism (CNN, DNN, and LSTM), the precision of CNN-ATT, DNN-ATT and LSTM-ATT has been improved by at least 15% and 10% on Google+ dataset and Sina Weibo dataset. This result shows that attention mechanism can better model the impact of intimacy difference between users and different friends on user representation. That is to say, CNN-ATT, DNN-ATT and LSTM-ATT assign different weights to different friends' ego-representation vector rather than the average strategy.

5.2 Occupation Predictions

Superiority of Mixed Data. Table 4 shows the performance of different models on user's occupation prediction task. Similar to gender prediction task, SVM is also the best method to use single data and LSTM is the best method to use mixed data. The performance of LSTM significantly outperforms SVM. It is reasonable to infer that LSTM captures more information to encode users' representation which is useful for enhancing the performance of occupation prediction. Similarly, the performance of CNN and DNN is not better than single data methods and the reason also relies in the problem of the long-term dependencies.

Table 4. Performance comparison between single and mixed data models

Model	Google+			Sina Weibo		
	Precision	Recall	F1	Precision	Recall	F1
CBOW	0.620	0.633	0.626	0.789	0.773	0.781
DeepWalk	0.643	0.658	0.650	0.807	0.812	0.809
Node2Vec	0.672	0.664	0.668	0.823	0.816	0.819
Skip-Gram	0.704	0.711	0.707	0.818	0.812	0.815
UE Model	0.719	0.709	0.714	0.842	0.853	0.847
Random Forest	0.730	0.724	0.727	0.863	0.859	0.861
SVM	**0.735**	**0.741**	**0.738**	**0.884**	**0.879**	**0.881**
CNN	0.601	0.614	0.607	0.751	0.735	0.743
DNN	0.688	0.668	0.678	0.816	0.804	0.810
LSTM	**0.831**	**0.837**	**0.834**	**0.918**	**0.897**	**0.907**

Effect of Attention Mechanisms. Table 5 shows the performance of models with hybrid data using attention mechanism. The F1 score of models which use attention mechanism is generally higher than that of models without attention mechanism. It has been improved by at least 10% and 8% respectively on Google+ and Sina Weibo. This result shows that different friend's representation has different contribute to the occupation prediction task and the attention mechanism can better model the impact of intimacy difference by assigning different weights to different friend's representation than the average strategy.

Table 5. Performance comparison among methods with attention mechanism

Model	Google+			Sina Weibo		
	Precision	Recall	F1	Precision	Recall	F1
CNN	0.601	0.614	0.607	0.751	0.735	0.743
CNN-ATT	**0.612**	**0.622**	**0.617**	**0.775**	**0.783**	**0.779**
DNN	0.688	0.668	0.678	0.816	0.804	0.810
DNN-ATT	**0.707**	**0.697**	**0.702**	**0.843**	**0.837**	**0.840**
LSTM	0.831	0.837	0.834	0.918	0.897	0.907
LSTM-ATT	**0.859**	**0.853**	**0.856**	**0.930**	**0.922**	**0.926**

5.3 Friend Recommendation

In this section, we compare the performance of single data (friendship) and mixed data (friendship and attributes) user representation methods on friend recommendation task. Generally speaking, the precision of these methods in friend recommendation task is lower than that in attribute prediction task. The cause lay in the fact that we delete some links when construct the social circle and the incompleteness of social links leads to the decrease of precision. But this does not prevent us from comparing the performance of these methods.

Table 6. Average accuracy comparisons of methods ($K = 10$)

Model	Google+	Sina Weibo
UE Model	0.706	0.725
Skip-Gram	0.711	0.732
Deep Walk	0.718	0.742
Node2Vec	0.724	0.754
CBOW	**0.737**	**0.763**
DNN	0.537	0.584
CNN	0.674	0.695
LSTM	**0.702**	**0.720**

Superiority of Mixed Data. From Table 6, CBOW is the best method to use single data and LSTM is the best method to use mixed data. In this task, the performance of the hybrid data methods is no better than that of the methods using only friendship data. The reason is that in the friend recommendation task, the similarity of user attributes is not the first basis for most users to consider when compared with existing friends. That is to say, most users expand their network of friends based on existing relationships. However, there are always some users who try to build new friends based on the similarity of their attributes. For these users, methods like CBOW that only use friendship information can not recommend friends very well. In this case, the mix data based methods (like LSTM, CNN and DNN) are obviously more suitable. In addition, as can be seen from Table 6, the performance of LSTM is closer to that of the methods using only friendship information, compared with CNN and DNN. The reason for this is that LSTM can capture richer temporal information.

Table 7. Average accuracy comparisons of methods with attention mechanism ($K = 10$)

Model	Google+	Sina Weibo
DNN	0.537	0.584
DNN-ATT	**0.556**	**0.608**
CNN	0.674	0.695
CNN-ATT	**0.687**	**0.704**
LSTM	0.702	0.720
LSTM-ATT	**0.718**	**0.738**

Effect of Attention Mechanism. From Table 7, we can see the effect of attention mechanism on friend recommendation task. The precision of CNN-ATT, DNN-ATT and LSTM-ATT has indeed improved. Especially LSTM-ATT, after using the attention mechanism, can more accurately adjust the weights of attribute information and friendship information, and then improve the precision of friend recommendation. It can be seen that the performance of LSTM-ATT is basically the same as that of the methods using only friendship information. At the same time, LSTM-ATT has the ability to recommend new friends to users according to their attribute similarity, which is impossible to use only friendship information.

6 Conclusion

In this paper, we proposed an unsupervised attribute-friendship attention framework for user representation, and design three concrete methods DNN-ATT, CNN-ATT and LSTM-ATT and validate the performance on three task: gender prediction, occupation prediction and friend recommendation. The experimental results show that the hybrid data model based on attention mechanism is effective and can model user representation more accurately and comprehensively.

In the future, we try to conduct more experiments on different data sets and introduce user posts in order to make user representations more informative and suitable for more tasks.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Benton, A., Arora, R., Dredze, M.: Learning multiview embeddings of Twitter users. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 14–19 (2016)
3. Chen, L., Qian, T., Zhu, P., You, Z.: Learning user embedding representation for gender prediction. In: 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 263–269. IEEE (2016)
4. Lefebvre-Brossard, A., Spaeth, A., Desmarais, M.C.: Encoding user as more than the sum of their parts: recurrent neural networks and word embedding for people-to-people recommendation. In: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, pp. 298–302. ACM (2017)
5. Liu, H., Wu, L., Zhang, D., Jian, M., Zhang, X.: Multi-perspective User2Vec: exploiting re-pin activity for user representation learning in content curation social network. Signal Process. **142**, 450–456 (2018)
6. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710. ACM (2014)
7. Song, Y., Lee, C.J.: Learning user embeddings from emails. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. vol. 2, pp. 733–738 (2017)
8. Tang, L., Liu, E.Y.: Joint user-entity representation learning for event recommendation in social network. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pp. 271–280. IEEE (2017)
9. Yu, J., Gao, M., Song, Y., Fang, Q., Rong, W., Xiong, Q.: Integrating user embedding and collaborative filtering for social recommendations. In: Romdhani, I., Shu, L., Takahiro, H., Zhou, Z., Gordon, T., Zeng, D. (eds.) CollaborateCom 2017. LNICST, vol. 252, pp. 470–479. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00916-8_44
10. Yu, Y., Wan, X., Zhou, X.: User embedding for scholarly microblog recommendation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 449–453 (2016)
11. Zeng, Z., Yin, Y., Song, Y., Zhang, M.: Socialized word embeddings. In: IJCAI, pp. 3915–3921 (2017)