Social Recommender System for GitHub using Network Embedding and Collaborative Filtering

1st Nisha C C
Department of Computer Science and Engineering
NSS College of Engineering
Palakkad, India
Email: nishacc93@gmail.com

2nd Anuraj Mohan
Department of Computer Science and Engineering
NSS College of Engineering
Palakkad, India
Email: anurajmohan@gmail.com

***Abstract*—GitHub is one of the most popular social coding environment on the Internet. Exploring the collaboration pattern and social network structure from the available GitHub data is helpful in solving several real-world tasks. Among them recommending relevant repositories for potential users has got greater attention in many works to improve the user experience in GitHub. One of the important directions towards the enhaned recommender result is to incorporate semantic social information along with the available user preference. Interestingly this direction should consider the issues of unavailability of explicit social information, difficulty in using social information directly into recommender systems and dealing with cold-start users who are cold at both rating and socialization. To address these issues we propose a model that utilizes semantic social information extracted using network embedding techniques into the recommender system. So we extend the Matrix Factorization framework with top-k semantic user information. Thus our proposed model enhances the recommendation result in an elegant manner.**

***Index Terms*—GitHub, Recommendation System, Network Representation Learning, Collaborative Filtering**

I. Introduction

GitHub[1] is one of the leading collaborative software artifacts on the Internet. It provides online code hosting platform with distributed version control (Git Revision Control) and social features (e.g., starring, following etc.) that support services ranging from finance to retail. As a collaborative social network, the platform provides public repositories to support the community of developers. They can create codes and push them to these repositories. The users of the GitHub repository can be a collaborator, a contributor or a follower. The collaborators of each repository have the rights to review the changes in the repository and can discard the changes if needed. Contributors can fork the repositories to work independently. Also, users can follow other users based on their interest.

In recent years, several pieces of research have been conducted with GitHub dataset available at GitHub Archive[2] and GHTorrent[3] to analyze the GitHub platform and its potential perils in terms of its social network structure and collaborative nature. Among them, an analysis conducted for GitHub open source survey dataset reveals that the collaboration and contribution statistics are spectacular in the year of 2017. In this year, the GitHub collaborative platform arrived with 24 million developers and 67 million repositories with 337 unique programming languages. This popularity and availability of metadata paid greater attention in GitHub data mining for applications like link prediction [13-16], recommendation [11-20], knowledge discovery [13], code reuse [6], and job profile creation [8]. Besides this, an interesting point of analysis shows that during the year of 2017 almost half of the persons signed up to the GitHub are students and totally new to programming. This highlights the role of GitHub as a learning platform.

The wide range of applicability and increased number of new users in the GitHub data motivated the researchers to work on building recommender system for finding relevant repositories to users [13-20], identifying potential contributors across the open source community [12], and detecting similar projects for code reuse and rapid prototyping [6]. Most of these research build recommender system using link prediction or Collaborative Filtering (CF) techniques. To enhance the recommender result recent works have focused on extracting social information. However, GitHub finds following challenges to build a recommender system. First, GitHub do not contain any explicit rating information. Second, the preferences of a user over the repository not only depend on the explicit or implicit feedback they provide but also the preferences of their social neighbors. Even though GitHub provides the characteristics of a social network, explicit and reliable social information are not always available. Third, in GitHub very active users need not be followed by many other users and may have different preferences. Hence, it is difficult to use social information directly into recommender systems. Finally, GitHub needs to deal with cold-start users who are cold at both rating and socialization.

To address these challenges we propose a social recommender system for GitHub that finds relevant repositories for potential users using Networks Representation Learning (NRL) and CF techniques. Through this proposed framework, we construct a bipartite network consist of a set

[1]https://github.com/
[2]https://www.githubarchive.org/
[3]http://ghtorrent.org/downloads.html

of users and repositories as nodes and interaction between them as edges. This network is projected to form a user-user interaction network to capture the social connections among the users. Then, inspired by NRL techniques, we make use of DeepWalk framework to generate latent feature representation for the nodes of the network. It will embed the nodes in a continuous vector space such that most similar node relies closely on the vector space. Then, top-k semantic users are extracted and this social information is incorporated into the Matrix Factorization (MF) framework for user preference prediction. Due to the unavailability of explicit rating information in GitHub data, we extract implicit feedback from users based on the number of times they interacted with the repositories. In summary, our main contributions are as follows:

- To incorporate social information into recommender system we make use of NRL techniques and finds top-k trustable social neighbors.
- We analyze the GitHub data available from GHTorrent to find implicit feedbacks of the user as the number of times the user interacted with the repositories including starring, following, commenting etc.
- We extend the Collaborative Filtering techniques for implicit feedback to leverage semantic social information.

The rest of this paper is organized as follows. Section 2 reviews several related works briefly. Section 3 introduces the proposed framework with possible enhancements. Finally, section 4 concludes with some notable insights and future scope of the proposed model.

II. RELATED WORKS

We classify related works into several categories. We first analyze the works related to GitHub data mining to identify potential problems, and shortcomings and to analyze the collaborative social structure of GitHub. Then we introduce several related works about building recommender system in GitHub. Also, we include some works that incorporate social information into recommender system. To enhance the recommendation result, we briefly discuss some NRL techniques that capture social information in an elegant manner. Finally, we review some recent works about recommender system that makes use of these network embedding techniques.

A. Study on GitHub Data

A number of works were analyzed GitHub data to derive some useful insights [1-10]. For example, E. Kalliamvakou, G. Gousios, K. Blincoe have made an empirical study about GitHub dataset with over 10.6 million repositories to take various potential perils into consideration[1]. This exploratory survey identifies possible threats to validity that are relative to the repository activity, users, and their characteristics. Also, the authors in [2] conducted a systematic analysis of the works on GitHub data and introduce some properties and improvements to enhance the quality of research that is focusing on GitHub. R. S. Geiger have done the recent statistical study for the 2017 GitHub survey data [3]. This summary analysis generates several reports that emphasize trends and patterns in the GitHub collaboration. Moreover, some authors [4] explored the characteristics of GitHub in terms of its social networks structure, collaborative nature and geographical impact on collaboration. Through this study the authors aim to give a starting point for the development of novel strategies in collaborative networks, in GitHub particularly, to improve the user experience.

Also, several research works have utilized the GitHub data to solve some real-world problems like finding socially-effective teams [5], detecting similar repositories using GitHub stars and README files [6], detecting communities in GitHub using weighted information from its social network structure [7], creating and suggesting matching developer profiles based on their programming skills and interests [8], and using GitHub as education platform for teachers and students [9]. We have analyzed all these works to create bigger insight into the importance and relevance of GitHub data analysis.

B. Recommender System for GitHub

Here, we are reviewing some recent works on building recommender system that identifies relevant contributors, reviewers or repositories for GitHub users. [11] introduces an automatic recommender system to predict reviewers for the incoming pull request. This method combines vector space model of information retrieval with semantic similarity score assignment of social network analysis. The authors in [12] identifies potential contributors to the given project using Weighted Collaborative Filtering(WCF) along with text matching-based recommendation algorithm. Another category of works uses link prediction techniques that infers some similarity metrics to find nearest neighbors [13, 16], Restricted Boltzmann Machine(RBM) for sparse feature generation [15], or other Machine Learning (ML) techniques [15] while building a recommender system. [17] combines Natural Language Processing (NLP) and ML techniques on README files to recommend relevant projects for users. The authors in [18] tries to rank the repositories based on the degree of correlation between given user and repositories present in the network structure constructed from historical GitHub data. The work in [19] introduces a multidimensional personalized recommendation system that combines different perspectives on project activities, task dependency of developers, and social mechanism in GitHub. By analyzing GitHub data and previous works, S. Sharma and A. Mahajan suggests possible approaches for building a recommender system for GitHub in in an elegant manner [20]. This includes all the techniques such as CF, content-based and hybrid-based recommendation [21-23].

From this review, it was found that while building a recommender system, most of the previous works ignores social network properties and collaborative patterns present in the

GitHub. Our proposed model tries to overcome this barrier by considering the trustable social connections of the users into the recommender system.

C. Recommender Systems with Social Information

CF techniques [21-23] are popular in recommender systems due to its versatility, cross-domain applicability, and large user-space support. It considers similar rating or ranking data as user preference for the items. An interesting form of data that identifies additional user preference is the social connections and is well-studied in recent years [24-27]. These works infer user preferences not only with the product rating, buying and viewing information but also with the preferences of their trustable social neighbors.

Recent trends in exploring social information leverage the notion of NRL techniques [28, 32-36]. All of these techniques are based on the assumption that the nodes of the network that relies closely on the embedded space will have the similar set of properties. B. Perozzi, R. Al-Rfou. and S.Skiena introduced the first and novel algorithm called DeepWalk [28] to learn the latent representation of the nodes of the network into low dimensional vector space. DeepWalk uses stochastic methods and generalizes natural language models [29-31] from the sequence of words to the network. To incorporate rich text information with the network structure, the authors in [32] explored the MF view of DeepWalk. LINE [33] is another framework introduced by J. Tang and M.Qu to embed large-scale information network preserving its first-order and second-order proximity. To learn latent representation for weighted graphs, S. Cao and W. Lu invented the GraRep [34] model and integrated the global network structure into the learning process. To preserve homophily and structural equivalence of the network the authors in [35] extended the DeepWalk algorithm and generated a semi-supervised algorithmic framework called Node2Vec. Instead of using shallow models [33], D. Wang, P. Cui, and W. Zhu introduced a semi-supervised SDNE [36] model that preserves first-order and second-order proximity using multiple non-linear functional units. All of these techniques are quite intuitive and fruitful in exploring social network properties. Hence many of the researchers reviewed the benefit of NRL techniques in an elegant manner [37, 38]. Different areas such as healthcare [39], social network analysis, and education [40] use these techniques to solve real-world problems.

Intuitively, recent works utilize network embedding techniques in recommender systems. The authors in [41] introduces recommender system for historical data based on the similarity of embedded vectors. [42] incorporate social information extracted through an extended DeepWalk model into the MF framework and emphasizes the presence of user-user trust information in recommender result. This work is closely related to our proposed model except they have worked with explicit rating information which is difficult to obtain from GitHub.

In summary, recommending relevant repositories for potential users in the GitHub is one of the study areas of current research works. Most of the proposals ignore social information generated from the GitHub social structure. Hence, we propose a model that identifies top-k similar users for a particular repository using the DeepWalk framework and extracts implicit feedback from the GitHub data thereby enhancing the recommendation result.

III. PROPOSED METHOD

In this section, we briefly introduce our proposed model for building a repository recommender system for GitHub. The model consists of a graph construction stage followed by an NRL stage to capture the semantic social information. Finally, we incorporate these social connections into MF framework to enhance the recommendation result. The general architecture is shown in the Fig.1.

A. Graph Construction

Initially, we construct a user-repository bipartite graph from the GitHub data available from GHTorrent. In this graph, a user node is connected with a repository node through the edge when the user interacted with the repository at least once. These interactions may be a push, comment, follow or star a repository. To leverage the social information using homogeneous network embedding techniques, we project the bipartite graph into user-user network $G = (V, E)$, where V is the set of users and $(u, v) \in E$ when the user u and v have at least one common repository in their interaction. Thus we constructed an undirected network without edge weights.

B. Extraction of Social Information using NRL

Inspired by text representation, Network Representation Learning aims to encode nodes of the network into a unified vector space with lower dimensionality. Given a network $G = (V, E)$, NRL finds a mapping function $\phi : v \in V \rightarrow \mathbb{R}^{|V| \times d}$, where $d << |V|$, such that every node $v \in V$ is mapped into a d-dimensional space preserving all the network properties(e.g., network connectivity, community structure, topology, etc.). There are a number of algorithms and methodologies to find appropriate mapping function.

Due to the scalability and performance, we use DeepWalk [28] for NRL and extract top-k semantic users from the constructed user-user network. DeepWalk is characterized by two procedures, a random walk that compute local community structure of the network and a SkipGram [30] that embed similarity of nodes into a continues vector space.

For the constructed network $G = (V, E)$ DeepWalk performs a truncated random walk W_{vi} rooted at node v_i such that for the random sequence $W_{v_i}^1, W_{v_i}^2, ..., W_{v_i}^k$ the node $W_{v_i}^{k+1}$ is chosen randomly from the neighbors of the node v_k.

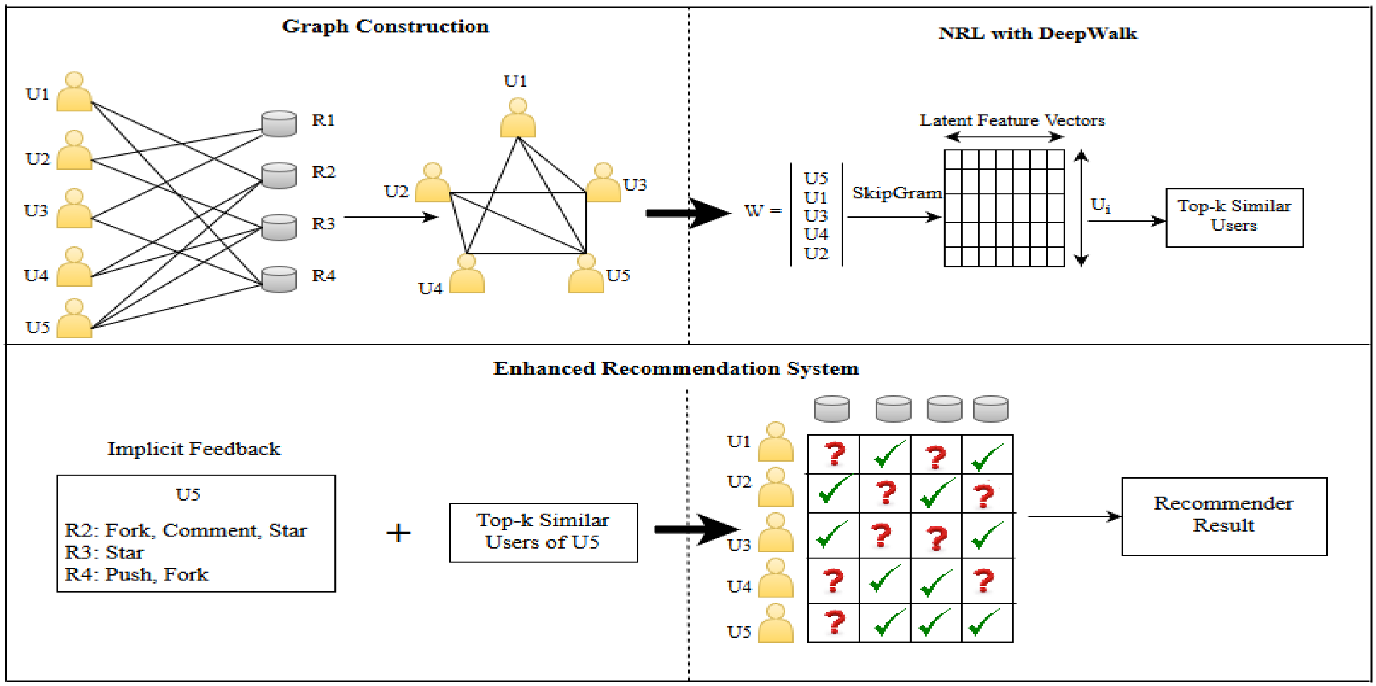

Fig. 1. Illustration of the proposed method. The first step is to project a user-user interaction network from the bipartite network constructed from GitHub dataset. Next step identifies top-k social neighbors using DeepWalk framework. Finally this social connections are incorporated with the feedback from the users(e.g., U5) to create an enhanced recommendation system.

It is observed that the frequency which nodes appear in the short random walks follows a power law distribution similar to the word frequency in natural language. Hence DeepWalk generalizes the SkipGram model from the sequence of words to the network.

Given a sequence of words $S = (W_0, W_1, ..., W_N)$, SkipGram defines the objective function:

$$\text{maximize} \quad \frac{1}{N}\sum_{t=1}^{N}\sum_{-w \leq t-t' \leq w} logP(W_t|W'_t) \tag{1}$$

where N is the vocabulary size and w is the window size for the context words. Interestingly, DeepWalk utilizes the random walk W_v into the SkipGram as the vocabulary set with walk length t as,

$$\text{maximize} \quad \sum_{v_t \in W_v}^{N}\sum_{-w \leq t-t' \leq w} logP(v'_t|\phi(v_t)) \tag{2}$$

where $P(v'_t|\phi(v_t))$is defined as a Softmax function and optimized with a Hierarchical Softmax to reduce the complexity of computation from $O(|V|)$ to $O(log|V|)$.

Finally, we compute the cosine similarity of embedded node vectors to identify their top-k similar nodes.

C. Repository Recommendation using Social Information

Next, we introduce the recommender system that incorporates social information from the user network. Here we address the challenge of unavailability of explicit rating information from GitHub by extracting implicit feedback. Most of the previous works make use of starring information for repository recommendation. But this information alone can provide an approximate level of interest of user over the repository. To address this issue we take all the type of interactions (e.g., follow, comment, push, etc.) into consideration.

Suppose the preference of m users with n repositories be an $m \times n$ matrix R, such that $r_{ij} \in R$ represents the number of activities performed by the user i with the repository j. Then we extend the MF framework for implicit feedback as follows. MF approximates $R \approx P^TQ$, where $P \in \mathbb{R}^{k\times m}$ and $Q \in \mathbb{R}^{k\times n}$ are latent features of users and repositories respectively. The preference of user i over the repository j is calculated as $\hat{r_{ij}} = p_i^T q_j$.

Incorporating semantic social information into MF framework yields the following objective function:

$$\begin{aligned} E = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}(p'_{ij} - p_i^T q_j)^2 + \frac{\lambda}{2}(\|P\|_F{}^2 + \|Q\|_F{}^2) \\ + \frac{\alpha}{2}\sum_{i=1}^{m}\sum_{f \in S(i)} \|p_i - p_f\|_F{}^2 \end{aligned} \tag{3}$$

where p'_{ij} is the preference indicator function given by,

$$p'_{ij} = \begin{cases} 1 & \text{if } r_{ij} > 0 \\ 0 & \text{if } r_{ij} = 0 \end{cases} \tag{4}$$

Also, if the user does not interact with a particular repository does not means that the user is not interested to work with that repository. As far as GitHub is concerned new users signing up to the GitHub may be a student or new to programming. So they may be unaware of the existence of these repositories in GitHub. Hence we use different confidence levels into the loss function as:

$$c_{ij} = 1 + \beta r_{ij} \tag{5}$$

This rate is controlled by β which is set to 40 for good results.

The model regularization parameter λ in (3) is used to avoid overfitting. $S(i)$ denotes top-k trustable social neighbors of user i. Then the function is optimized using Stochastic Gradient Descent(SGD) with following gradients w.r.t p_i and q_j.

$$\begin{aligned} \frac{\partial E}{\partial p_i} = c_{ij}(p'_{ij} - p_i^T q_j)q_j + \lambda p_i + \alpha \Bigg[\sum_{f \in S(i)} (p_i - p_f) \\ - \sum_{g \in \tilde{S(i)}} (p_g - p_i) \Bigg] \end{aligned} \tag{6}$$

$$\frac{\partial E}{\partial q_j} = c_{ij}(p'_{ij} - p_i^T q_j)p_i + \lambda q_j \tag{7}$$

By optimizing this objective function we have arrived at an improved result for the recommendation system in GitHub.

IV. CONCLUSION

We have reviewed different milestones in GitHub data analysis to identify requirements needed to build a repository recommender system for GitHub. Almost all the previous works[11-20] builds such recommender system either using link prediction or other similarity measures and ignores the social mechanism provided by GitHub. Some recent works show that it is possible to improve the recommendation result using semantic social information. We have seen that the advancements in NRL techniques can better capture the social structure of the GitHub. Hence we propose a model that not only takes the user preferences for the recommendation but also incorporate user-user trust information. Apart from the previous works in GitHub we have utilized two main features. First, social connections are extracted using an NRL framework called DeepWalk. Second, rather than using an approximate level of user preference from repository starring we include all the type of interactions of the users in the preference level. Thereby, we have enhanced the CF techniques to get an improved result.

In future, we plan to extend this recommender system with user and repository attributes. Also, instead of using homogenous network embedding techniques, we plan to use a reliable heterogeneous network embedding technique with some specific features to handle bipartite networks which are inevitable in the recommendation.

ACKNOWLEDGMENT

We are thankful to the Department of Computer Science and Engineering at N.S.S. College of Engineering, Palakkad for providing required facilities.

REFERENCES

[1] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian. "An in-depth study of the promises and perils of mining GitHub." *Empirical Software Engineering*, pages 1-37, 2015.

[2] V. Cosentino, J. Luis, and J. Cabot. "Findings from GitHub: methods, datasets and limitations." *in MSR 2016: Proceedings of the 13th Inter national Workshop on Mining Software Repositories*. ACM, pages 137-141, 2016.

[3] R. S. Geiger. "Summary analysis of the 2017 github open source survey." *ArXiv*:1706.02777, 2017.

[4] A. Lima, L. Rossi, and M. Musolesi. "Coding together at scale: Github as a collaborative social network," *in CoRR*, abs/1407.2535, 2014.

[5] A. Majumder, S. Datta, and K. Naidu. "Capacitated team formation problem on social networks." *in SIGKDD*, pages 1005-1013, 2012.

[6] Yun Zhang, David Lo, Pavneet Singh Kochhar, Xin Xia, Jianlai Li, and Jianling Sun. "Detecting similar repositories on GitHub," *in SANER*, pages 13-23, 2017.

[7] Joseph Mc Arrama, T. L. "Social coding: Evaluating githubs network using weighted community detection." *CS224W Project Report*, 2012.

[8] Claudia Hauff and Georgios Gousios. "Matching GitHub developer profiles to job advertisements." *in Proceedings of the 12th Working Conference on Mining Software Repositories*, pages 362-366. IEEE Press, 2015.

[9] Zagalsky, A., Feliciano, J., Storey, M. A., Zhao, Y., and Wang W. "The Emergence of GitHub as a collaborative platform for education." *in Proceedings of CSCW* , pp. 1906-1927, 2015.

[10] F. Thung, T. Bissyande, D. Lo, and L. Jiang. "Network structure of social coding in github. *in 17th European Conference on Software Maintenance and Reengineering (CSMR)*, pages 323-326, 2013.

[11] Y. Yu, H. Wang, G. Yin, and C. Ling. "Reviewer recommender of pull requests in GitHub." *in ICSME*, pages 609-612. IEEE, 2014.

[12] X. Zhang, T. Wang, G. Yin, C. Yang, and H. Wang. " Who will be interested in? A contributor recommendation approach for open source projects." *in SEKE*, 2017.

[13] D. Schall. "Link prediction in directed social networks." *Social Netw. Analys. Mining*, 2014.

[14] R. Bartusiak, T. Kajdanowicz, A. Wierzbicki, L. Bukowski, O. Jarczyk, and K. Pawlak. "Cooperation prediction in github developers network with restricted boltzmann machine." *in Intelligent Information and Database Systems. Springer Berlin Heidelberg*, pages 96-107, 2016.

[15] Radoslaw Nielek et al. "Choose a job you love: Predicting choices of GitHub developers." *In IEEE/WIC/ACM WI*, Omaha, NE, USA, pages 200-207, 2016.

[16] T. Matek and S. T. Zebec. "GitHub open source project recommendation system."*in CoRR*, vol. abs/1602.02594, 2016.

[17] R. L. Q. Portugal, M. A. Casanova, T. Li, and J. C. S. do Prado Leite. "GH4RE: Repository recommendation on GitHub for requirements elicitation reuse." *in CAiSE*, 2017.

[18] C Yang, Q Fan, T Wang, G Yin, and H Wang. "RepoLike: Personal repositories recommendation in social coding communities." *in Proceedings of the 8th Asia-Pacific Symposium on Internet ware*, pages 54-62, 2016.

[19] Yang et al. "RepoLike: a multi-feature-based personalized recommendation approach for open source repositories." *in FITEE*, 2017.

[20] S Sharma and A Mahajan. "Suggestive approaches to create a recommender system for GitHub." *in IJITCS*, 2017.

[21] D Das, L Sahoo, S Datta. "A Survey on recommendation system." *in IJCA*, vol. 160, 2017.

[22] Y.F. Hu, Y. Koren, and C. Volinsky. "Collaborative filtering for implicit feedback datasets." *in Proc. IEEE International Confonference on Data Mining (ICDM 08)*, pages 263-272, 2008.

[23] X. Yang, Y. Guo, Y. Liu, and H. Steck. "A survey of collaborative filtering based social recommender systems." *Computer Communications*, vol. 41, pages 1-10, 2014.
[24] A. J. Chaney, D. M. Blei, and T. Eliassi-Rad. "A probabilistic model for using social networks in personalized item recommendation." *in RecSys*, 2015.
[25] G. Guo, J. Zhang, and N. Yorke-Smith. "Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings." *in AAAI*, 2015.
[26] B. Yang, Y. Lei, D. Liu, and J. Liu. "Social collaborative filtering by trust." in IJCAI, 2013.
[27] T. Zhao, J. McAuley, and I. King. "Leveraging social connections to improve personalized ranking for collaborative filtering." *in CIKM*, 2014.
[28] B. Perozzi, R.Al-Rfou. and S.Skiena. "DeepWalk: online learning of social representations." *in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-KDD 14*, pages 701-710, 2014.
[29] Ferrone, L. and Zanzotto, F. M. "Symbolic, distributed and distributional representations for natural language processing in the era of deep learning: A survey." *ArXiv*, pages 1-25, 2017.
[30] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean. "Distributed representations of words and phrases and their compositionality." *Advances in Neural Information Processing Systems*, pages 3111-3119, 2013.
[31] X. Rong. "word2vec parameter learning explained." *ArXiv*, pages 1-19, 2014.
[32] C. Yang, Z. Liu, D. Zhao, M. Sun and E.Y. Chang. "Network representation learning with rich text information." *in Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 2111-2117, 2015.
[33] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. "Line: Large-scale information network embedding." *Proceedings of the 24th International Conference on World Wide Web*, pages 1067-1077, 2015.
[34] S. Cao, W. Lu, and Q. Xu. "GraRep: Learning graph representations with global structural information." *in Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891-900, 2015.
[35] A. Grover, and J. Leskovec. "node2vec: Scalable feature learning for networks." *in Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855-864, 2016.
[36] D. Wang, P. Cui, and W. Zhu. "Structural deep network embedding." *in Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
[37] G. Zhong, L. N. Wang, and J. Dong. "An overview on data representation learning: From traditional feature learning to recent deep learning." *ArXiv*:1611.08331, 2016.
[38] Moyano, L. G. "Learning network representations A review with applications to complex networks." *The European Physical Journal Special Topics, 226*, pages 499518, 2017.
[39] Z. Jin, R. Liu, Q. Li et. al. "Predicting user's multi-interests with network embedding in health-related topics." *Proceedings of the International Joint Conference on Neural Networks*, pages 2568-2575, 2016.
[40] Y. Chen,B. Perozzi, and S. Steven. "Vector-based similarity measurements for historical figures" *in JIS*, pages 163-174, 2016.
[41] W. Zhao, J. Huang, and J. Wen. "Learning distributed representations for recommender systems with a network embedding approach." *in AIRS*, pages 224-236, 2016.
[42] L. Yu, and Y. Wang. "Collaborative user network embedding for social recommender systems", *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 381-389, 2017.