



Intra-view and Inter-view Attention for Multi-view Network Embedding

Yueyang Wang¹(✉), Liang Hu², Yueting Zhuang¹, and Fei Wu¹

¹ College of Computer Science and Technology,
Zhejiang University, Hangzhou, China
{yueyangw,yzhuang,wufei}@zju.edu.cn

² School of Mechanical Engineering, Zhejiang University, Hangzhou, China
h111325003@zju.edu.cn

Abstract. Network Embedding, which represents nodes in networks with efficient low-dimensional vectors, has been proved useful in a variety of applications. However, most existing approaches study single-view networks but not the multi-view networks with multiple types of relationships between nodes. Meanwhile, they ignore the rich features associated with the nodes, which is common in real world. In this paper, we propose a novel network embedding method, Intra-view and Inter-view attention for Multi-view Network Embedding (I^2 MNE), which leverages both the multi-view network structure and the node features to efficiently generate node representations. Specially, we introduce the intra-view attention when aggregating node features from neighbors for each single view and the inter-view attention when integrating representations across different views. Experiments on two real-world networks show that our approach outperforms other counterpart network embedding methods.

Keywords: Multi-view network · Network embedding
Network representation learning · Attention

1 Introduction

With the development of Internet, information networks are ubiquitous in the real world (e.g. academic networks [21,22], movie recommendation networks [18]). These information networks have attracted deep insight and one of the most popular analysis methods is the network embedding, i.e., network representation learning. Network embedding embeds the networks into the low-dimensional space to directly measure the neighborhood similarities between different nodes and has been proved to be highly effective when solving the traditional tasks such as node classification [2], clustering [3], link prediction [1].

Traditional network embedding approaches (e.g. DeepWalk [15], node2vec [6], LINE [24]) mainly focus on the representation learning for networks with a single type of relationship. Later, researchers further propose some convolutional neural network embedding methods (e.g. GCN [8], graphSAGE [7]) to encode

both network structure and features of nodes. However, the majority of practical information networks are *de facto* multi-view networks, involving multiple types of relationships and node features. Figure 1 shows an example of multi-view network. In this academic network, relationships of authors may include co-author relationship which means whether two authors have been collaborated on a paper and citing relationships which means whether one author cited the papers written by the other one. Features of authors may include the research interests and titles of papers they write.

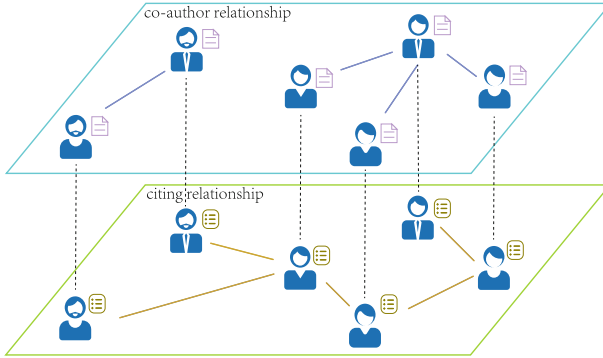


Fig. 1. An example of multi-view network which contains multiple types of relationships between nodes and different node features in each single view.

Recently, the multi-view network embedding problems has received more attention. Various methods proposed for learning representations with multiple views perform well on many applications [4]. Nevertheless, they suffer from specific limitations as well. Methods [5, 10, 20] based on matrix factorization are faced with the expensive computational cost, thus are not suitable for large-scale data. Clustering methods [12, 27, 28] neglect the different importance of different views with a lack of consideration about weight learning. And more recent methods like [16, 19] based on deep learning miss the utilization of node features which promote the embedding performance besides the network structure.

In this paper, we propose Intra-view and Inter-view attention for Multi-view Network Embedding (I^2MNE), a novel method to overcome the limitations mentioned above. The intra-view attention is introduced to specify the different importance of neighbors when aggregating neighbor features of each node for single view. Similarly, the inter-view attention is also introduced to assign the different significance of views when integrating representations across different views. The attention weights and the representations can be efficiently trained through the back propagation algorithm.

We conduct experiments on two real-world datasets of different domains. The effectiveness of our approach is evaluated on the classification task. The experimental results show that I^2MNE outperforms other state-of-the-art methods. To summarize, we make the following contributions:

- We propose I²MNE to study multi-view network embedding, which aims to learn node representations by leveraging structure and feature information from multiple views.
- We introduce intra-view attention when aggregating the node features from neighbors and inter-view attention when integrating representations across different views to learn robust node representations.
- We conduct experiments on two real-world multi-view networks. Experimental results demonstrate the effectiveness and efficiency of our proposed approach over many competitive methods.

The rest of this paper is organized as follows. In Sect. 2, we describe the problem definition. In Sect. 3, we present the I²MNE algorithm for multi-view network embedding in detail. In Sect. 4, we analyze the learned node representation and compare the proposed model with the existing network embedding approaches on two real world datasets. Conclusions are given at the end.

2 Problem Definition

In this section, we formally define the problem of network embedding in multi-view network. Firstly, the multi-view network is defined as follows:

Definition 1 (Multi-view Network). *A multi-view network is defined as $G = (V, E, C)$, where V is the set of nodes representing objects; $E = \cup_{k=1}^K E_k$, K is the number of views, $E_k \subseteq V \times V$ is the set of edges representing relationships between two nodes in view k and $C = \cup_{k=1}^K C_k$, $C_k = \{\mathbf{f}_i, \forall n_i \in V\}$ denotes the features of objects in view k , $\mathbf{f}_i \in \mathbb{R}^F$ denotes features of node n_i and F is the dimension of features.*

Our goal is to learn the node representations in the multi-view network. We define this problem as follows:

Definition 2 (Multi-view Network Embedding). *Given a multi-view network, denoted as $G = (V, E, C)$, the aim of multi-view network embedding is to learn low-dimensional representations $O \in \mathbb{R}^{|V| \times d}$, where $d \ll |V|$ is the number of embedding dimensions.*

3 Our Approach

In this section, we present our approach, I²MNE, which embeds the nodes with features in multi-view network into a common space. We first aggregate the features of each node’s neighbors for single view to encode the node proximities and node features. Then we integrate node representations across different views. Inspired by the recent progress of the attention mechanism [11], we introduce the intra-view attention (shown in Fig. 2) to automatically specify different weights to nodes within neighbors and inter-view attention (shown in Fig. 3) to assign different weights to views.

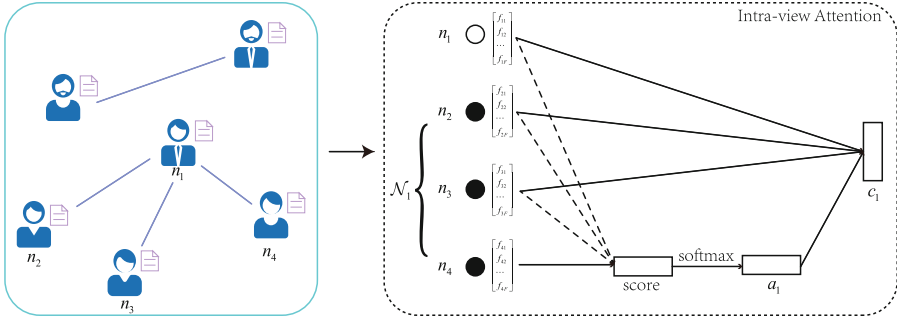


Fig. 2. The Intra-view Attention. It specifies different importance to neighbors ($n_2, n_3, n_4 \in \mathcal{N}_1$ in this example) of each node (n_1 in this example).

3.1 Embedding Generation

Intra-view Attention. In each single view, for each node n_i with its feature \mathbf{f}_i and its neighbors' features $\mathbf{F}_i \in \mathbb{R}^{|\mathcal{N}_i| \times F}$, where \mathcal{N}_i denotes the neighbors of n_i , we first introduce a content-based score function as follow:

$$\mathbf{s}_i = \text{score}(\mathbf{f}_i, \mathbf{F}_i) = \mathbf{f}_i \mathbf{W}_a \mathbf{F}_i^T \tag{1}$$

where $\mathbf{W}_a \in \mathbb{R}^{F \times F}$ is the intra-view attention weight matrix. This function indicates the importance of neighbors' features to node n_i . To make the scores comparable across different nodes, we use a softmax function to normalize them as follow:

$$[\mathbf{a}_i]_j = \text{softmax}_j(\mathbf{s}_i) = \frac{\exp([\mathbf{s}_i]_j)}{\sum_{j'=1}^{|\mathcal{N}_i|} \exp([\mathbf{s}_i]_{j'})} \tag{2}$$

where $[\cdot]_j$ means the j -th value of the vector and $j \in \{1, \dots, |\mathcal{N}_i|\}$.

Then, we introduce the context-vector \mathbf{c}_i which captures the relevance of features between n_i and its neighbors \mathcal{N}_i using the normalized attention scores as follow:

$$\mathbf{c}_i = \mathbf{a}_i \mathbf{F}_i \tag{3}$$

Next, we use a weight matrix $\mathbf{W} \in \mathbb{R}^{F' \times F'}$, where F' is the dimension of hidden layer, to get the aggregated vector as follow:

$$\mathbf{f}'_i = \sigma(\mathbf{W}[\mathbf{f}_i \oplus \mathbf{f}_{\mathcal{N}_i} \oplus \mathbf{c}_i]) \tag{4}$$

where $\mathbf{f}_{\mathcal{N}_i}$ is the feature vector aggregated by the neighbors of n_i , σ is the activation function and \oplus is the concatenation operation. And the aggregation strategy of $\mathbf{f}_{\mathcal{N}_i}$ can be chosen as [7].

Finally, we apply the aggregated vector with a normalization to get the hidden representation \mathbf{z}_i and search P depth for neighbor sampling [7]. The details can be found in Algorithm 1 step 1–12.

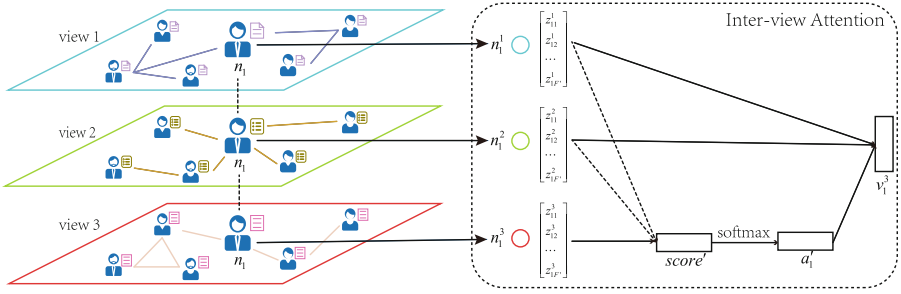


Fig. 3. Inter-view Attention. It assigns different importance to hidden representations from different views.

Inter-view Attention. For each \mathbf{z}_i^k of view k , where $k \in \{1, 2, \dots, K\}$, we introduce inter-view attention as follow:

$$\mathbf{s}'_i^k = \text{score}'(\mathbf{z}_i^k, \bar{\mathbf{Z}}_i^k) = \mathbf{z}_i^k \mathbf{W}_{a'}^k \bar{\mathbf{Z}}_i^k \mathbf{T} \quad (5)$$

$$[\mathbf{a}'_k]_j = \text{softmax}_j(\mathbf{s}'_i^k) = \frac{\exp([\mathbf{s}'_i^k]_j)}{\sum_{j'=1}^{K-1} \exp([\mathbf{s}'_i^k]_{j'})} \quad (6)$$

where $\bar{\mathbf{Z}}_i^k$ is the set of all the hidden representations for node n_i except view k , $j \in \{1, \dots, K-1\}$ and $\mathbf{W}_{a'}^k \in \mathbb{R}^{F' \times F'}$ is the inter-view attention weight matrix.

Then, the view-context vector \mathbf{v}_i^k is defined to integrate the hidden representations from different views as follow:

$$\mathbf{v}_i^k = \mathbf{a}'_k \bar{\mathbf{Z}}_i^k \quad (7)$$

Finally, the final node representation is defined as follow:

$$\mathbf{o}_i = \sigma(\mathbf{W}'[\mathbf{Z}_i \oplus \mathbf{V}_i]) \quad (8)$$

where $\mathbf{Z}_i = [\mathbf{z}_i^1 \oplus \mathbf{z}_i^2 \oplus \dots \oplus \mathbf{z}_i^K]$, $\mathbf{V}_i = [\mathbf{v}_i^1 \oplus \mathbf{v}_i^2 \oplus \dots \oplus \mathbf{v}_i^K]$ and $\mathbf{W}' \in \mathbb{R}^{2KF' \times d}$ is the weight matrix. The details can be found in Algorithm 1 step 13–14.

3.2 Parameters Learning of I²MNE

For each single view, we apply a graph-based loss function with negative sampling techniques [13, 14] to $\mathbf{z}_i, \forall n_i \in V$ as follow:

$$L_s(\mathbf{z}_i) = -\log(\sigma(\mathbf{z}_i^T \mathbf{z}_j)) - \sum_{n=1}^N E_{v_n \sim P_n(n_j)} \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_{v_n})) \quad (9)$$

where n_j is a node that co-occurs near n_i on fixed-length random walk, v_n is a negative sample, $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function, N is the number of negative samples and P_n denotes the negative sampling distribution.

Algorithm 1. I²MNE

Input: A multi-view network $G = (V, E, C)$; depth P ; non-linearity σ ; differentiable aggregate functions $g_p, \forall p \in \{1, \dots, P\}$; neighborhood function \mathcal{N}

Output: the node representations $\mathbf{o}_i \in O$ where $O \in \mathbb{R}^{|V| \times d}$

```

1: for  $k = 1$  to  $K$  do
2:   for  $p = 1$  to  $P$  do
3:      $\mathbf{f}_i^0 \leftarrow \mathbf{f}_i, \forall n_i \in V$ 
4:     for  $n_i \in V$  do
5:        $\mathbf{f}_{\mathcal{N}_i}^p \leftarrow g_p(\{\mathbf{f}_j^{p-1}, \forall n_j \in \mathcal{N}_i\})$ 
6:       calculate the context-vector  $\mathbf{c}_i^p$  according to Eq.(3)
7:       calculate the  $\mathbf{f}'_i^p$  according to Eq.(4)
8:     end for
9:      $\mathbf{f}_i^p \leftarrow \mathbf{f}_i^p / \|\mathbf{f}_i^p\|_2, \forall n_i \in V$ 
10:    end for
11:     $\mathbf{z}_i^k \leftarrow \mathbf{f}_i^P, \forall n_i \in V$ 
12:  end for
13: calculate the view-context vector  $\mathbf{v}_i^k, \forall k \in \{1, \dots, K\}, \forall n_i \in V$  according to Eq.(7)
14: calculate the final node representation  $\mathbf{o}_i, \forall n_i \in V$  according to Eq.(8)

```

This loss function encourages that the embeddings of connected nodes are similar to each other, while enforcing that the embeddings of disparate nodes are highly distinct.

For multi-views integration, we apply cross-entropy loss for the classification task as follow:

$$L_m = \sum_{n_i \in S} L(\mathbf{z}_i, y_i) \quad (10)$$

where S is the set of labeled nodes, y_i is the label of node n_i , and L is the cross-entropy loss function.

With the above definitions, the overall loss function is defined as follows:

$$L = \sum_{n_i \in V} L_s(\mathbf{z}_i) + L_m \quad (11)$$

Our objective is to minimize the overall loss function and it can be efficiently optimized with the back propagation algorithm [17]. Following the suggestion of [16], in each iteration of the optimization, we first optimize graph-based loss on each single view, learn the hidden representations and tune the parameters of intra-view attention. Then update the parameters of inter-view attention with the labeled data by optimizing the cross-entropy loss.

4 Experiment

4.1 Datasets

The detailed statistics of the datasets are shown in Table 1.

Table 1. Statistics of datasets

Dataset	# of nodes	# of edges in each view		Features in each view		# of labels
AMiner	16,604	62,115 (co-author)	266,802 (citing)	interest (co-author)	title (citing)	8
Flickr	35,314	3,017,530 (friendship)	3,496,495 (tag-sim)	- (friendship)	tag (tag-sim)	171

AMiner. We use AMiner dataset [25]¹ to analyze the research fields of authors. There are two views in the multi-view author network including the co-author relationship view and the citing relationship view. The former is constructed by authors who publish the same paper and the latter is constructed by authors who cite others’ paper. The features of authors in the co-author view are the research interests. We use word2vec [13] pre-trained by English Wikipedia Skip-Gram² to learn the textual features. And the titles of all the papers published by each author are treated as the node features in the citing view. we use Doc2vec [9] pre-trained by English Wikipedia DBOW⁴ to learn the textual features. We choose all the papers from the most popular venue³ in eight research fields defined by [23] and select all the relative authors who publish these papers. There are 16604 authors with labels in the filtered dataset.

Flickr. We use Flickr dataset [26]⁴ to analyze the community membership of users. The multi-view user network includes the friendship view and the tag-similarity view. For the latter one, we first calculate the user similarity by their tags using TF-IDF. Then we construct the tag-similarity relationship using 100-nearest neighbors. And the textual features are the tags of users learned by word2vec.

4.2 Compared Algorithms

In this section, our proposed approach is compared with the following methods for performance analysis:

Single-view Methods:

- LINE [24]: A network embedding method without node features.
- Deepwalk/node2vec [6, 15]: We find that the results of them have no significant differences, so we use $p = 1$ and $q = 1$ [6] in node2vec for comparison.
- GraphSAGE [7]: A network embedding method with node features.
- I²MNE-Intra: A variant of our proposed method only use the intra-view attention for single views.

¹ <https://aminer.org/aminernetwork>.

² <https://github.com/jhlau/doc2vec>.

³ 1. IEEE Trans. Parallel Distrib. Syst. 2. STOC 3. IEEE Communications Magazine

4. ACM Trans. Graph. 5. CHI 6. ACL 7. CVPR 8. WWW.

⁴ <http://dmml.asu.edu/users/xufei/datasets.html>.

Multi-view Methods

- *-concat: We concatenate the embeddings learned from all the single views by the node2vec, GraphSAGE and I²MNE, respectively.
- *-mean: We calculate the mean average of the embeddings learned from all the single views by the node2vec and GraphSAGE, respectively.
- MVE [16]: A multi-view network embedding method with attention mechanism, which can also apply to our problem but cannot utilize the features of nodes and the attention is different from ours.
- I²MNE-NoInter: A variant of our proposed method without the inter-view attention and calculate the mean average embeddings of all the single views.
- I²MNE: our proposed method for multi-view network embedding with node features.

4.3 Parameter Settings

For all the methods except *-concat, the dimension is set as 128 by default. For the concatenation methods, the dimension is set as $128K$, where K is the number of views. The dimension of features is set as 300. The number of negative samples is set as 5, and the learning rate is set as 0.001. For node2vec, we set the walk length as 40, the window size as 10. All the embedding vectors are finally normalized.

4.4 Results

We evaluate the network embeddings on the classification task. A logistic regression classifier fed by the embeddings of all labeled nodes is employed. We set 75% nodes as training data and the rest of nodes are used for testing. The classification experiments are repeated independently for 10 times and the averaged Macro-F1 and Micro-F1 measures are reported in Table 2. Note that, for single-view methods, the best results on single views are reported. From this table, we have the following observations:

- (1) For the single view, I²MNE-Intra achieves the best performance than the other single-view methods. In addition, I²MNE-concat also outperforms the other concatenation methods for the multi-view network. These indicate that the intra-view attention can capture the impact of neighbors' features on the nodes and improve the performance.
- (2) On the Flickr dataset, I²MNE achieves significant improvement comparing to all the methods on both measurements. I²MNE-NoInter already outperforms all the baselines in terms of Macro-F1 due to the importance of intra-view attention. And I²MNE gains further improvement over I²MNE-NoInter with 0.0534 Macro-F1 score. It shows that the inter-view attention plays an important role in our methods.

- (3) Especially for the AMiner dataset, I²MNE achieves significant improvements than other methods with 0.0768 \sim 0.1125 gains in terms of Macro-F1. In addition, we observe that I²MNE achieves the best performance in terms of Macro-F1 while I²MNE-NoInter gets the best results in terms of Micro-F1. It is perhaps that the classes in the AMiner dataset are imbalance, so the Macro-F1 which treats all the classes equally is more reasonable than the Micro-F1 which equally treats all the instances. From the result, we also see that the inter-view attention can preserve more distinction between classes compared to the intra-view attention.

Table 2. Results of classification on both datasets.

Category	Algorithm	AMiner		Flickr	
		Macro-F1	Micro-F1	Macro-F1	Micro-F1
Single view	LINE	0.4219	0.5804	0.5108	0.8438
	Deepwalk/node2vec	0.5494	0.6878	0.4937	0.8524
	GraphSAGE	0.6134	0.7533	0.5462	0.8971
	I ² MNE-Intra	0.6380	0.7608	0.5650	0.8996
Multi view	node2vec-concat	0.5739	0.7601	0.4981	0.8342
	GraphSAGE-concat	0.6095	0.8036	0.5602	0.8942
	I ² MNE-concat	0.6116	0.8055	0.5748	0.8973
	node2vec-mean	0.5884	0.7860	0.4938	0.8507
	GraphSAGE-mean	0.5786	0.7950	0.5134	0.8923
	MVE	0.5527	0.7544	0.4919	0.8836
	I ² MNE-NoInter	0.5735	0.7977	0.5340	0.8896
	I ² MNE	0.6652	0.7769	0.5874	0.8962

5 Conclusion

In this paper, we propose two effective attentions to learn representations of multi-view networks associated with node features, named I²MNE. We introduce the intra-view attention to leverage the feature information from neighbors and the inter-view attention to make full use of the information from different views. Experiments on two real-world datasets demonstrate that our proposed model is effective and efficient for multi-view network embedding. In the future, we plan to apply our model for more tasks (e.g. link prediction). What’s more, we also plan to investigate the embedding of networks with edge features.

Acknowledgments. This work is partially supported by the National Natural Science Foundation of China (Nos. U1509206, 61625107, U1611461), the Key Program of Zhejiang Province, China (No. 2015C01027).

References

1. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: ACM International Conference on Web Search and Data Mining, pp. 635–644 (2011)
2. Bhagat, S., Cormode, G., Muthukrishnan, S.: Node classification in social networks. *Comput. Sci.* **16**(3), 115–148 (2012)
3. Ding, C.H.Q., He, X., Zha, H., Gu, M., Simon, H.D.: A min-max cut algorithm for graph partitioning and data clustering. In: IEEE International Conference on Data Mining, pp. 107–114 (2001)
4. Elkahky, A.M., Song, Y., He, X.: A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: International Conference on World Wide Web, pp. 278–288 (2015)
5. Greene, D.: A matrix factorization approach for integrating multiple data views. In: European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 423–438 (2009)
6. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 855 (2016)
7. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, pp. 1024–1034 (2017)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2016)
9. Lau, J.H., Baldwin, T.: An empirical evaluation of doc2vec with practical insights into document embedding generation (2016)
10. Liu, J., Wang, C., Gao, J., Han, J.: Multi-view clustering via joint nonnegative matrix factorization (2013)
11. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421 (2015)
12. Ma, G., et al.: Multi-view clustering with graph embedding for connectome analysis. In: ACM on Conference on Information and Knowledge Management, pp. 127–136 (2017)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: International Conference on Neural Information Processing Systems, pp. 3111–3119 (2013)
14. Mnih, A., Teh, Y.W.: A fast and simple algorithm for training neural probabilistic language models. In: International Conference on International Conference on Machine Learning, pp. 419–426 (2012)
15. Perozzi, B., Alrfou, R., Skiena, S.: DeepWalk: online learning of social representations, pp. 701–710 (2014)
16. Qu, M., Tang, J., Shang, J., Ren, X., Zhang, M., Han, J.: An attention-based collaboration framework for multi-view network representation learning, pp. 1767–1776 (2017)
17. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* **323**(6088), 399–421 (1986)
18. Shi, C., Zhang, Z., Luo, P., Yue, Y., Yue, Y., Wu, B.: Semantic path based personalized recommendation on weighted heterogeneous information networks. In: ACM International on Conference on Information and Knowledge Management, pp. 453–462 (2015)

19. Shi, Y., Han, F., He, X., Yang, C., Luo, J., Han, J.: mvn2vec: preservation and collaboration in multi-view network embedding (2018)
20. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 650–658 (2008)
21. Sun, Y., Barber, R., Gupta, M., Aggarwal, C.C., Han, J.: Co-author relationship prediction in heterogeneous bibliographic networks. In: International Conference on Advances in Social Networks Analysis and Mining, pp. 121–128 (2011)
22. Sun, Y., Han, J., Aggarwal, C.C., Chawla, N.V.: When will it happen?: relationship prediction in heterogeneous information networks, pp. 663–672 (2012)
23. Swami, A., Swami, A., Swami, A.: metapath2vec: scalable representation learning for heterogeneous networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 135–144 (2017)
24. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: International Conference on World Wide Web (2015)
25. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: ArnetMiner: extraction and mining of academic social networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 990–998 (2008)
26. Wang, X., Tang, L., Liu, H., Wang, L.: Learning with multi-resolution overlapping communities. *Knowl. Inf. Syst.* **36**(2), 517–535 (2013)
27. Xia, T., Tao, D., Mei, T., Zhang, Y.: Multiview spectral embedding. *IEEE Trans. Syst. Man Cybern. Part B* **40**(6), 1438–1446 (2010)
28. Zhou, D., Burges, C.J.C.: Spectral clustering and transductive learning with multiple views. In: Proceedings of the Twenty-Fourth International Conference on Machine Learning, pp. 1159–1166 (2007)