Hindawi
BioMed Research International
Volume 2020, Article ID 6248686, 9 pages
https://doi.org/10.1155/2020/6248686

Research Article

PmDNE: Prediction of miRNA-Disease Association Based on Network Embedding and Network Similarity Analysis

Junyi Li,[1] **Ying Liu**,[1] **Zhongqing Zhang**,[1] **Bo Liu**,[2] **and Yadong Wang**[1,2]

[1] *School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, Guangdong 518055, China*
[2] *School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China*

Correspondence should be addressed to Junyi Li; lijunyi@hit.edu.cn and Yadong Wang; ydwang@hit.edu.cn

Received 24 July 2020; Revised 20 October 2020; Accepted 21 October 2020; Published 7 December 2020

Academic Editor: Zhenguo Zhang

Successful prediction of miRNA-disease association is nontrivial for the diagnosis and prognosis of genetic diseases. There are many methods to predict miRNA and disease, but biological data are numerous and complex, and they often exist in the form of network. How to accurately use the features of miRNA and disease-related biological networks to predict unknown association has always been a challenge. Here, we propose PmDNE, a method based on network embedding and network similarity analysis, to predict the miRNA-disease association. In PmDNE, the structure of network bipartite graph is improved, and a random walk generator is designed. For embedded vectors, 128 dimensions are used, and the accuracy of prediction is significantly improved. Compared with other network embedding methods, PmDNE is comparable and competitive with the state of art methods. Our method can solve the problem of feature extraction, reduce the dimension of features, and improve the efficiency of miRNA-disease association prediction. This method can also be extended to other area for biomedical network prediction.

1. Introduction

microRNA (miRNA) is a kind of noncoding RNA with length of around 22 nucleotides. It has been found in plants, animals, and viruses. Recent studies have shown that microRNAs play an important role in different biological processes [1]. It is able to prevent tumor invasion, control cell growth, regulate cell cycle regulation, and so on. Studies have also shown that many miRNAs are involved in human diseases [2], such as cancer, viral diseases, and immune-related diseases [3–5]. Therefore, successful prediction of disease-related miRNAs is nontrivial for the diagnosis and prognosis of genetic diseases and drug development.

How to predict human miRNA in the relationship and make good use of the existing miRNA disease association data is an important topic in the study of human diseases. For biomedicine, the accuracy of data is very important. There are many public databases related to miRNAs such as mir2disease [6], miRBase [7], and TarBase [8]. With the increasing concern of the scientific community on the relationship between diseases and miRNAs, their data are also included. For instance, HMDD [9] is the miRNA human disease association database established in 2007.

There are two kinds of major methods to predict disease-miRNA association. The first method is based on traditional network iteration, and the second one is based on machine learning.

In the traditional iterative method, the miRNAs and the nodes in the disease network are iterated, and the possible relationship is found from high to low through the final convergence result ranking. In 2016, Chen et al. suggested that global network similarity can capture the association between disease and miRNA more effectively than traditional local network similarity. Therefore, RWRMDA [10] method was developed to predict potential miRNA-disease associations. Chen et al. also proposed a computational model of HGIMDA [11], which integrates the known miRNA disease association, different types of disease similarity, and miRNA similarity into the heterograph to predict new disease-related miRNAs. However, the method of using network has its own disadvantages. It may be biased towards the well-known miRNAs and diseases. In the network method, restarting random walk is very time-consuming and parameters with transition probability; different selection of parameters also affects the final results. The experimental results of this kind of method highly depend on the reliable biological network model and cannot be applied to new miRNAs or new diseases.

The second kind of method is based on machine learning. This kind of method is able to solve the problem of new miRNAs and disease relation prediction. In 2011, Xu introduced a method [12] based on miRNA target imbalance network (MTDN) to give priority to new disease-related miRNAs. A weighted KNN-based HDMP [13] method is proposed by Xuan et al. In addition, the semantic similarity and phenotypic similarity of diseases are used to calculate the functional similarity matrix of miRNA. Chen proposed a semisupervised learning RLSMDA [14] model to predict potential disease-related miRNAs in 2014. RLSMDA [14] can calculate miRNA disease association prediction score of new diseases. This kind of method needs to solve two major problems: feature extraction and negative case missing.

Recently, people pay more and more attention to the network embedding method [15, 16]. It extracts features by extracting some relations of complex data and embeds the high latitude features of complex data into low dimensional space. In order to better predict the relationship between disease and miRNA, the network embedding method can be used to solve the problem of feature extraction. Therefore, we propose a method based on network embedding and network similarity analysis called PmDNE to predict the miRNA-disease association. In PmDNE, the structure of network bipartite graph is improved, and a random walk generator is designed. The accuracy of prediction has improved. Compared with other network embedding methods, PmDNE is comparable and competitive with the state of art methods. Our method can solve the problem of feature extraction, reduce the dimension of features, and improve the efficiency of miRNA-disease association prediction. This method can also be extended to other area for biomedical network prediction.

2. Materials and Methods

2.1. miRNA Disease Association Data. miRNA disease association data is obtained from the database HMDD3.0 (http://www.cuilab.cn/hmdde). In order to predict the effect effectively, we use the latest version of HMDD3.0. Some other databases, such as mri2disease, are not up to date. Some databases do not focus on the relationship between miRNA and disease, so we chose HMDD. A total of 894 disease nodes and 1208 miRNA nodes are obtained from the HMDD database, and 18733 diseases and miRNA association relationships are obtained as shown in Tables 1 and 2.

2.2. Disease Similarity Data about the Disease Similarity Network. We construct a directed acyclic graph (DAG) to describe the disease according to the literature [17] of Wang et al. Based on the medical subject title descriptor, it can be downloaded from the national medical library (http://www.nlm.nih.gov/). A total of 414003 related disease similarity relationships were obtained as shown in Table 2.

Table 1: Number of edges about miRNA and disease, miRNA and miRNA, and disease and disease.

	miRNA	Disease
miRNA	644918	18733
Disease	18733	414003

Table 2: The number of miRNA nodes and disease node.

	Nodes number
miRNA	1208
Disease	894

2.3. miRNA Similarity Data. miRNA similarity network is based on the method of calculating miRNA functional similarity proposed by Wang et al. [17]. The functional similarity of 495 miRNA nodes was obtained by downloading miRNA function similarity data conveniently.

2.4. Isomorphic Network Construction and Binetwork Construction. When constructing miRNA-disease binetwork, if there is correlation, the weight of their edges is 1, and the weight of nonexistent edges is 0. In this way, we can transform the prediction method into a binary classification problem. For isomorphic network, the weight of similarity data is set as the weight of isomorphic network.

2.5. The First Similarity Obtains the Node Embedding Vector of Graph Representation Learning. In order to better reconstruct the original network in the low dimensional space after embedding, the first similarity relation is represented by the existing edge learning, and the second similarity relationship is represented by the edge learning with transitive relationship. The final node representation is learned by combining the two methods. The modelling of explicit relationship is the same as the first similarity of Line [18]. By considering local similarity, the compactness of two connected nodes is defined as Equation (1).

$$(i, j) = \frac{W_{ij}}{\sum e_{ij} \in E w_{ij}}, \tag{1}$$

where w_{ij} is the edge e_{ij}'s weight. The denominator is the sum of the weights of all edges. If two nodes are linked together, the probability of two nodes appearing together after embedding is very high.

Many research works [19, 20] have achieved good results on measuring the similarity of two nodes embedded in the space. Most of them refer to the idea of taking vector inner product of word2vec [21]. Herein, we also use this method to define the possibility of two nodes adjacent

```
Algorithm: WalkGenerator(W, R, maxT, minT, p, c, e).
Input: weight maxtrix of the bipartite network W, vertex set R, times of max walks from per vertex maxT, times of min walks per vertex minT, walk stopping probability p,the weight of disease or miRNA's similarity network c and e。.
Output: a set of vertex sequences $D^R$
calculate vertices' centrality:H=CentralityMeasure(W);
calculate $W^R$ by Equation (4);
foreach vertex $v_i$ do
    l=max(H($v_i$)*maxT, minT);
    for i=0 to l do
        $D_{v_i}$=BiasedRandomWalk(W, $v_i$,p);
        Add $D_{v_i}$ into $D^R$
Return $D^R$
```

Pseudocode 1: Pseudocode of node sequence.

Table 3: Concept of TF, FN, FP, and TN.

Prediction values	Actual values	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

PR curve: abscissa is recall rate and ordinate is precision; precision = TP/(TP + FP); recall = TP/(total positive samples) = TP/(TP + FN); ROC curve: the abscissa is FPR and the ordinate is TPR; TPR = TP/(TP + FN); FPR = FP/(TN + FP).

to each other in the embedded space as Equation (2).

$$\widehat{P}(i,j) = \frac{1}{1 + \exp\left(-\overrightarrow{U_i}^{T} \overrightarrow{V_j}\right)}, \tag{2}$$

where $u_i v_j$ is the embedded node vector. Embedding vector means minimizing the difference between two nodes. That is, the closer the original node is, the closer the embedded node is still the closest relationship. To minimize the difference of the possibility of appearing together before and after embedding, KL divergence is used.

$$\begin{aligned} \text{Minimize } O_1 = \mathrm{KL}\left(P \,|\, |\widehat{P}\right) &= \sum_{e_{ij}\in E} P(i,j) \log\left(\frac{P(i,j)}{\widehat{P}(I,J)}\right) \\ &= \propto - \sum_{e_{ij}\in E} w_{ij} \log \widehat{P}(i,j). \end{aligned} \tag{3}$$

The equation above represents that the closer the distance between the two nodes before and after embedding, the smaller the KL divergence is, the more similar the two distributions are. The local information of the original network is retained through the first similarity relationship. In other words, for two closely connected nodes, the representation of the two nodes learned in this way is also close to each other in the low dimensional vector space.

2.6. The Second Similarity Obtains the Node Embedding Vector of Graph Representation Learning. We model the second similarity relationship and extract the feature vector by DeepWalk [20]. But the feature vectors embedded by the random walk of DeepWalk are all based on the same type of nodes. Therefore, we embed the nodes based on such a theory. Although there are no directly connected edges between two nodes of the same type, if there is a path from u_i to u_j, it can be considered that there is a relationship between the two nodes. If two nodes of the same type are connected to the same node, they can be considered as having links. In PmDNE, we need to split a bipartite graph into two homogeneous networks, and combine it with miRNA network similarity and disease semantic similarity network. Through Equations (4) and (5), we generate two corpora containing different types of nodes. Then the random walk model is used to determine the node sequence library, and skipgram is used to obtain the similarity feature vector.

$$W_{ij}^{\ D} = \sum_{k\in m} w_{ik} w_{jk} + c d_{ij}, \tag{4}$$

$$W_{ij}^{\ M} = \sum_{k\in d} w_{ik} w_{jk} + e m_{ij}, \tag{5}$$

where $w_{ik} w_{jk}$ is the weight from i to j and j to k and d_{ij} is the weight of nodes i to j in the disease similarity network. m_{ij} is the weight of nodes i to j in miRNA network. c and e are the weights of similarity networks.

However, the random walk strategy of DeepWalk is not optimal, so we redesign a random walk method. The specific way is as follows:

(1) Obtain two networks with the same type of nodes by Equations (4) and (5) and construct two homogeneous networks by combining disease semantic similarity and miRNA functional similarity

(2) The more links for one node, the more important the proof is, and the more random walk sequences start from it

(3) Many random walk strategies [22] are to produce fixed length sequences, which does not conform to the actual rule of node embedding. The number of words in each sentence is uncertain. Therefore, we obtain node sequences of different lengths by making random walk stop or return to the original initial

TABLE 4: Influence of different networks on results.

	ROC_AUC	PR_AUC	PREC	ACC	F1	Recall
1	0.8952 ± 0.003	0.9002 ± 0.002	0.6744 ± 0.01	0.8153 ± 0.02	0.8104 ± 0.004	0.7863 ± 0.004
2	0.8833 ± 0.002	0.8916 ± 0.0015	0.6480 ± 0.015	0.8034 ± 0.02	0.7986 ± 0.004	0.7861 ± 0.005
3	0.8906 ± 0.0015	0.8966 ± 0.002	0.663 ± 0.001	0.8103 ± 0.015	0.8054 ± 0.003	0.7857 ± 0.004
4	0.8914 ± 0.001	0.8968 ± 0.0015	0.6634 ± 0.003	0.8115 ± 0.02	0.8056 ± 0.002	0.7813 ± 0.004

node at a certain step. The algorithm of measuring node importance we can chooses centrality algorithm or hits [23]. Pseudocode 1 shows the pseudocode of node sequence obtained by random walk.

Then, skipgram [24] algorithm is used to learn the embedded vector.

$$\text{Maximize } O_2 = \prod_{u_i \in S \wedge S \in D^U} \prod_{u_C \in C_s(u_i)} p(u_c \mid u_i), \tag{6}$$

$$\text{Maximize } O_3 = \prod_{v_j \in S \wedge S \in D^v} \prod_{v_C \in C_s(v_j)} p(v_c \mid v_j), \tag{7}$$

where $p(u_c \mid u_i)$ softmax is used for output.

$$p(u_c \mid u_i) = \frac{\exp\left(\vec{u_i}^T \vec{\theta_c}\right)}{\sum_{k=1}^{|U|} \exp\left(\vec{u_i}^T \vec{\theta_k}\right)}, \tag{8}$$

$$p(v_c \mid v_j) = \frac{\exp\left(\vec{v_j}^T \vec{\theta_c}\right)}{\sum_{k=1}^{|V|} \exp\left(\vec{v_j}^T \vec{\theta_k}\right)}. \tag{9}$$

However, due to the large amount of denominator calculation of softmax, we adopt the method of negative sampling [24–26], which transforms the calculation of each context vector into a binary classification problem of noncontext vector and context vector.

$$p\left(u_c, N_S^{NS}(u_i) \mid u_i\right) = \prod_{z \in u_c \wedge N_S^{NS}(u_i)} p(z \mid u_i), \tag{10}$$

$$p(z \mid u_i) = \begin{cases} \sigma\left(\left(\vec{u_i}^T \vec{\theta_z}\right)\right) & \text{if } z \text{ is } u_i'\text{s context} \\ 1 - \sigma\left(\vec{u_i}^T \vec{\theta_c}\right) & z \in N_S^{NS}(u_i) \end{cases}. \tag{11}$$

2.7. Obtain the Node Embedding Vector of the Final Graph Representation Learning. The function formula of the final optimization is Equation (12).

$$\text{Maximize } L = \alpha \log O_2 + \beta \log O_3 - \gamma O_1. \tag{12}$$

In the end, the embedding vector is obtained by iterating the embedding vector with random gradient descent [27]. For example, we use random gradient descent to update $\vec{u_i}$ and $\vec{v_j}$ for O_1 :

$$\vec{u_i} = \vec{u_i} + \lambda\left\{\gamma w_{ij}\left[1 - \sigma\left(\vec{u_i}^T \vec{v_j}\right)\right] * \vec{v_j}\right\}, \tag{13}$$

$$\vec{v_j} = \vec{v_j} + \lambda\left\{\gamma w_{ij}\left[1 - \sigma\left(\vec{u_i}^T \vec{v_j}\right)\right] * \vec{u_i}\right\}, \tag{14}$$

where σ is the sigmoid function and λ is the learning rate.

For O_2 and O_3, gradient descent is also used to update $\vec{u_i}$ and $\vec{v_j}$:

$$\vec{u_i} = \vec{u_i} + \lambda\left\{\sum_{z \in \{u_c\} \cup N_S^{NS}(u_i)} \alpha\left[I(z, u_i) - \sigma\left(\vec{u_i}^T \vec{\theta_z}\right)\right] * \vec{\theta_z}\right\}, \tag{15}$$

$$\vec{v_j} = \vec{v_j} + \lambda\left\{\sum_{z \in \{v_c\} \cup N_S^{NS}(v_j)} \beta\left[I(z, v_j) - \sigma\left(\vec{v_j}^T \vec{\vartheta_z}\right)\right] * \vec{\vartheta_z}\right\}, \tag{16}$$

where $I(z, u_i)$ is i in u_i's context, if exist $I(z, u_i)$ is 1 and 0 if not. $I(z, v_j)$ is similarity. For the centre word's contextual and noncontext word vectors $\vec{\theta_z}$ and $\vec{\vartheta_z}$, they are defined as (17) and (18).

$$\vec{\theta_z} = \vec{\theta_z} + \lambda\left\{\alpha\left[I(z, u_i) - \sigma\left(\vec{u_i}^T \vec{\theta_z}\right)\right] * \vec{u_i}\right\}, \tag{17}$$

$$\vec{\vartheta_z} = \vec{\vartheta_z} + \lambda\left\{\beta\left[I(z, v_j) - \sigma\left(\vec{v_j}^T \vec{\vartheta_z}\right)\right] * \vec{v_j}\right\}. \tag{18}$$

By considering both the first similarity relation and the second similarity relation, the node representation is learned. Then, we can use random forests to make predictions.

2.8. Criteria for Validation of Prediction. For the binary classification problem, according to the combination of real class and learner prediction category, the examples can be divided into true positive example (TP), false negative example (FN), false positive example (FP), and true negative example (TN) as shown in Table 3.

AUC is the area under the curve, and its calculation method takes into account the classification ability of the

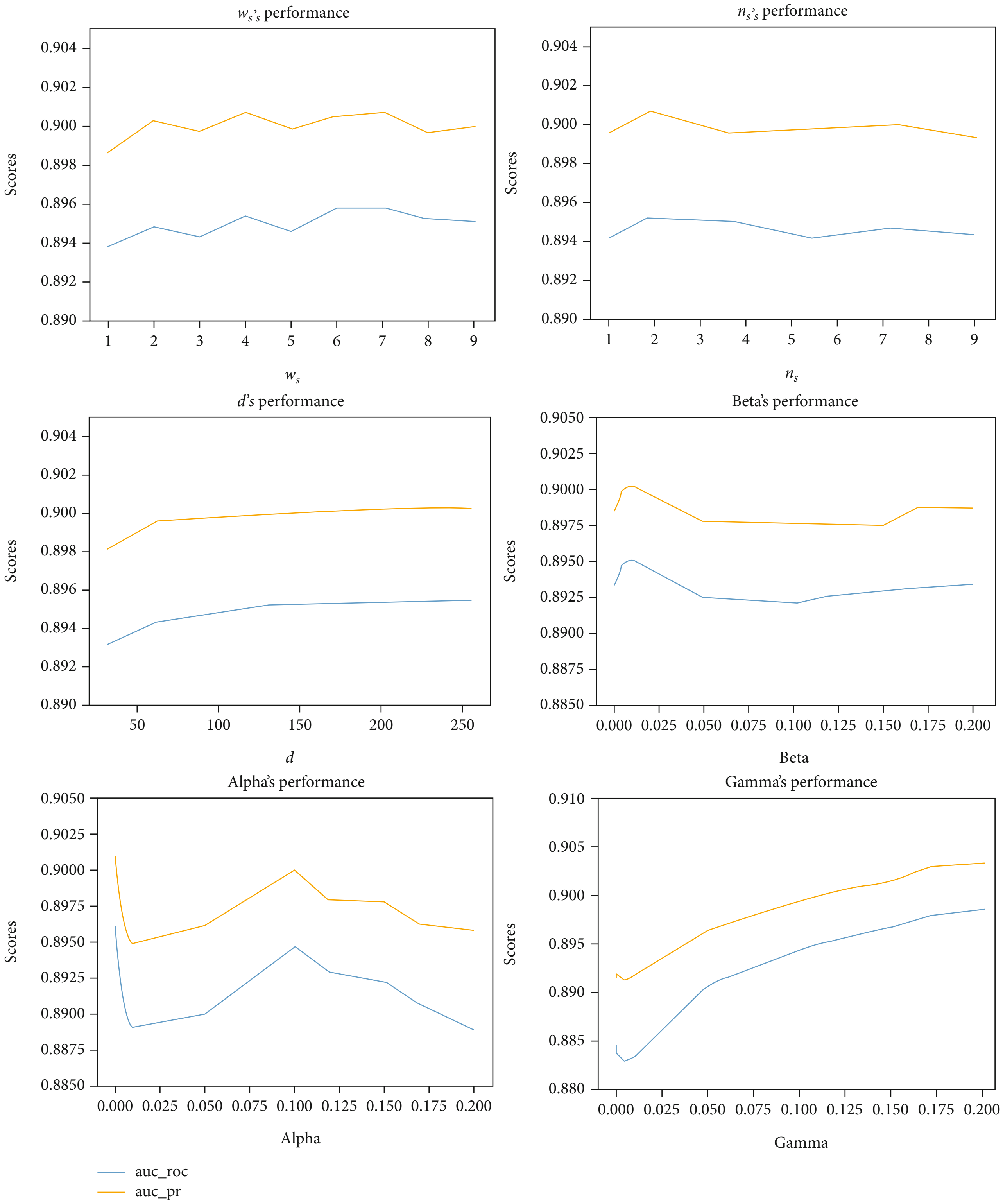


Figure 1: Influence of parameters on prediction effect. The parameter scores mean the value obtained by ROC or PR. The scores of alpha, beta, and gamma fluctuate greatly. These three parameters play an important role in regulating the size of the first similarity and the second similarity.

Table 5: Comparison of network embedding methods.

	Auc_roc	Auc_pr
PmDNE	0.8954 ± 0.003	0.9002 ± 0.002
DeepWalk	0.8689 ± 0.002	0.8780 ± 0.002
Line	0.8302 ± 0.003	0.8305 ± 0.002
Node2Vec	0.8807 ± 0.004	0.8782 ± 0.004
GraRep	0.8766 ± 0.002	0.8760 ± 0.003
GF	0.8881 ± 0.004	0.8856 ± 0.003
Lap	0.7706 ± 0.004	0.7062 ± 0.002
lle	0.8670 ± 0.004	0.8673 ± 0.004

classifier for positive and negative cases. In the case of unbalanced samples, the classifier can still make a reasonable evaluation. The larger the AUC, the more advanced the prediction results of the samples and the better the prediction effect. In addition to the above two important indicators, we also select precision and accuracy; F1 scores and recall were used as the evaluation criteria.

3. Results and Discussion

3.1. Results. The 4 : 1 data set is divided into training set and test set, and the final features are obtained by five cross validation. The feature dimension is 128 dimensions, and 2102 node vectors are obtained. Because this is an unbalanced classification task, we solve this problem by randomly selecting the same number of unconnected edges as negative examples. The random forest [28] is used to predict the parameters. For the weight of the similarity between the two networks, we choose 0.5 that is half. The maximum number of steps max_t of random walk is 32, and the minimum number of steps is 1, 0.15 for the probability of stopping immediately. 0.0001, 0.01, and 0.1 are for the three optimization objective functions, respectively. The AUC values of ROC and PR are 0.8954 ± 0.001 and 0.9002 ± 0.0015.

We also measure the results of adding network similarity and not adding network similarity. The results shown in Table 4 are as follows: 1 is the embedding method with two similar networks added, 2 is the embedding method without adding network, 3 is the embedding method with adding disease network, and 4 is the result of adding miRNA similarity network. From the results, we can see the result of adding similar network. It is the best. This shows that we have greatly improved the prediction effect by adding network similarity.

3.2. Computational Efficiency. Because this paper uses Python implementation, so the time efficiency will be lower than other embedding methods completed by C++. C++ is closer to the bottom, so the efficiency will be improved. However, Python has many data processing-related libraries, which will make the code writing more convenient. In this paper, the running time efficiency is minute level, and other methods are seconds' level.

3.3. Parameter Analysis. Important parameters are analyzed as shown in Figure 1. The parameter scores mean the value obtained by ROC or PR, and w_s is the size of the context window after selecting a central word in the random walk corpus. As the window becomes larger, the AUC of ROC and AUC of PR increase, which is in line with the actual law. The more context is, the more accurate the prediction will be. However, when the window reaches a certain value, AUC of ROC and AUC of PR tend to be stable, because the context information is enough to produce prediction results. n_s is the number of negative samples selected. With the increase of the number of negative samples, the prediction will be more accurate. d is the dimension of the embedded vector. It can be concluded that the higher the dimension is, the more original information it retains, and the more accurate the prediction is. But when it reaches a certain value, it also tends to be stable. α, β, and γ are the coefficients of the optimization function, respectively. It can be seen that the fluctuation is very obvious, which indicates that they are important parameters to balance the first similarity and the second similarity for the embedded vector. Single increase of explicit or implicit relationship will lead to the uneven proportion of the first similarity and the second similarity, which will lead to the fluctuation of the prediction results.

3.4. Comparison of Network Embedding Methods. In order to compare the characteristics of this paper, we select six methods, such as DeepWalk, line, node2vec, grarep [29], GF, lap [30], and LLE [31], and we compare the results. The same data and prediction methods are used to measure the performance of this method. The following are the introduction of some of these methods and the results of comparative experiments.

DeepWalk: a node embedding method for heterogeneous networks, which obtains node sequences through unbalanced random walks, and then uses word2vec to obtain embedding vectors

Line: by optimizing the first and second similarity in a heterogeneous network, the final node vector is obtained

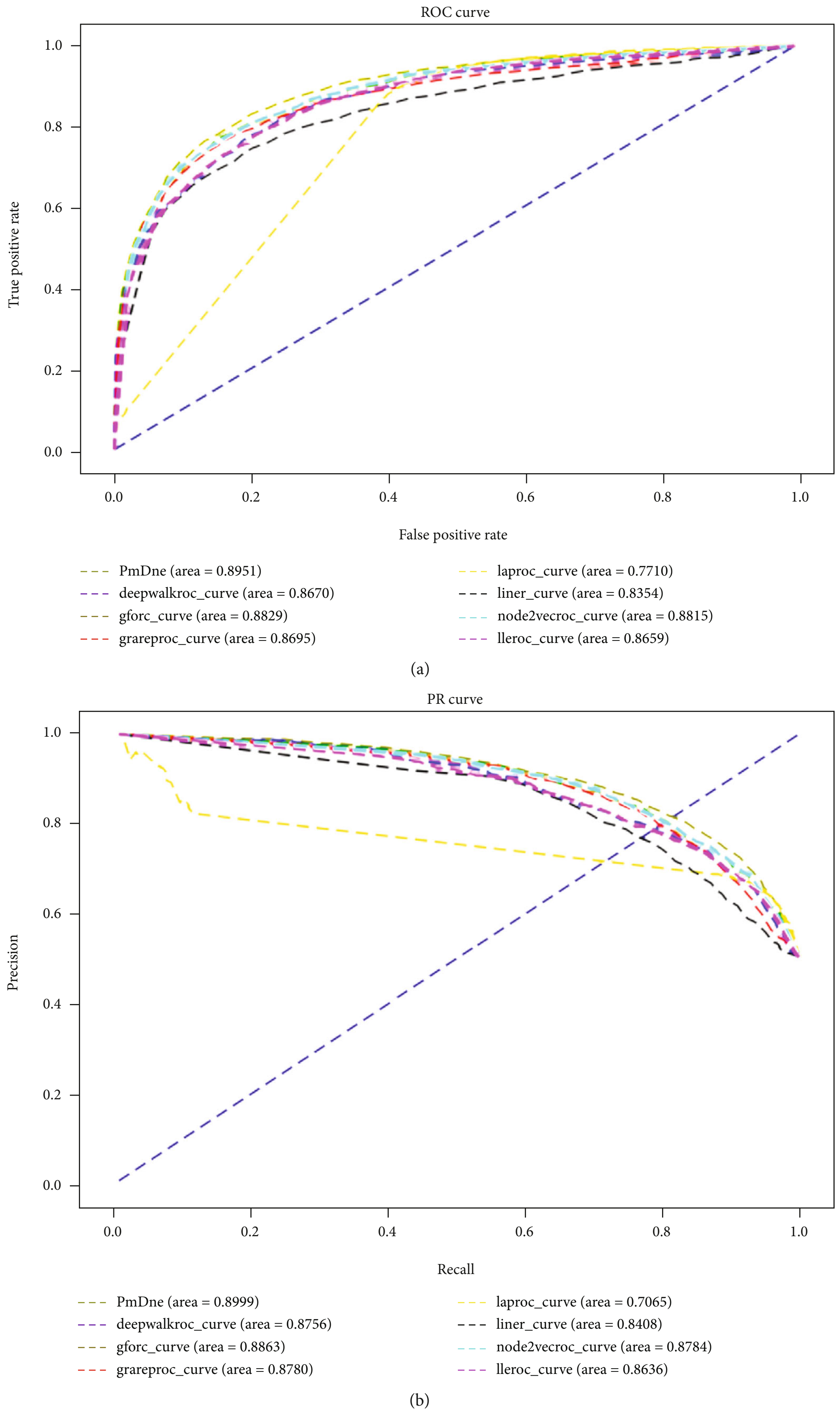
Node2vec: inherits DeepWalk and generates node sequence through organized random walk

Grarep: using matrix decomposition to solve network embedding problem. It can deal with weighted networks and integrate the global structure information in the process of learning network representation. However, due to the large amount of computation, this method will be particularly time-consuming, so it cannot be used in large-scale networks

GF: higher order nearest neighbor keeps embedding. By introducing higher order similarity matrix, higher-order similarity is preserved by generalized singular value decomposition to obtain embedding vector

From Table 5 to Figure 2, we can see that the ROC and AUC of PR method in this paper are better than other network embedding methods.

3.5. Comparison of Different Classifiers. Table 6 shows the comparison of the prediction results of different classifiers on the embedded vector. We compared six classifiers. The



(a)

(b)

Figure 2: Experimental results for PR and ROC curves of each models: (a) ROC curves for all models; (b) PR curves for all models.

Table 6: Comparison of the effect of different classifiers.

	ROC_AUC	PR_AUC	PREC	ACC	F1	Recall
RF	0.8954 ± 0.003	0.9002 ± 0.002	0.6744 ± 0.01	0.8153 ± 0.02	0.8104 ± 0.004	0.7863 ± 0.004
KNN	0.8746 ± 0.002	0.8560 ± 0.0015	0.7933 ± 0.015	0.8075 ± 0.02	0.8014 ± 0.004	0.7758 ± 0.005
GBC	0.8827 ± 0.0015	0.8955 ± 0.002	0.7747 ± 0.001	0.8045 ± 0.015	0.7937 ± 0.003	0.7532 ± 0.004
SVM	0.7693 ± 0.001	0.8194 ± 0.0015	0.7367 ± 0.003	0.7042 ± 0.02	0.5989 ± 0.002	0.4495 ± 0.004
LR	0.8070 ± 0.02	0.8412 ± 0.005	0.7541 ± 0.004	0.7390 ± 0.015	0.7153 ± 0.004	0.65618 ± 0.005
ADBC	0.8330 ± 0.002	0.8579 ± 0.002	0.71370.0015	0.7570 ± 0.002	0.7348 ± 0.004	0.6757 ± 0.005

RF is the Random Forest Classifier; KNN is the K Neighbors Classifier; ADBC is the AdaBoost Classifier; LR is the Logistic Regression Classifier; GBC is the Gradient Boosting Classifier; SVM is the support vector machines.

4. Conclusion

We propose PmDNE, a method based on network embedding and network similarity analysis, to predict the miRNA-disease association. For embedded vectors, 128 dimensions are used, and the accuracy of prediction is significantly improved. The values of PR and AUC of PmDNE are 0.9002 and 0.8954, respectively. Compared with other network embedding methods, PmDNE has better ability on extract the features of disease and miRNA networks. Our method improves the efficiency of miRNA-disease association prediction. This method can also be extended to other area for biomedical network prediction.

Data Availability

miRNA disease association data is obtained from the database website: HMDD3.0, link: http://www.cuilab.cn/hmdde. Disease theme data is downloaded from the national medical library, link: http://www.nlm.nih.gov. The functional similarity of 495 miRNA nodes was obtained by downloading miRNA function similarity data [17].

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Authors' Contributions

JL and YL designed the study, performed bioinformatics analysis, and drafted the manuscript. All of the authors performed the analysis and participated in the revision of the manuscript. JL and YW conceived the study, participated in its design and coordination, and drafted the manuscript. All authors read and approved the final manuscript.

Acknowledgments

This work was supported by the grants from the National "863" Key Basic Research Development Program (2014AA021505), the National Key Research Program (2017YFC1201201), the Shenzhen stable support program, and the startup grant of Harbin Institute of Technology (Shenzhen).

Supplementary Materials

Supplementary file 1: main code for PmDNA. *(Supplementary Materials)*

References

[1] D. P. Bartel, "microRNAs: target recognition and regulatory functions," *Cell*, vol. 136, no. 2, pp. 215–233, 2009.

[2] N. Meola, V. A. Gennarino, and S. Banfi, "microRNAs and genetic diseases," *Pathogenetics*, vol. 2, no. 1, p. 7, 2009.

[3] B. J. Reinhart, F. J. Slack, M. Basson et al., "The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans," *Nature*, vol. 403, no. 6772, pp. 901–906, 2000.

[4] J. Brennecke, D. R. Hipfner, A. Stark, R. B. Russell, and S. M. Cohen, "bantam encodes a developmentally regulated microRNA that controls cell proliferation and regulates the proapoptotic gene hid in Drosophila," *Cell*, vol. 113, no. 1, pp. 25–36, 2003.

[5] E. A. Miska, "How microRNAs control cell division, differentiation and death," *Current Opinion in Genetics & Development*, vol. 15, no. 5, pp. 563–568, 2005.

[6] Q. Jiang, Y. Wang, Y. Hao et al., "miR2Disease: a manually curated database for microRNA deregulation in human disease," *Nucleic Acids Research*, vol. 37, suppl_1, pp. D98–D104, 2008.

[7] S. Griffiths-Jones, H. K. Saini, S. van Dongen, and A. J. Enright, "miRBase: tools for microRNA genomics," *Nucleic Acids Research*, vol. 36, no. Database, pp. D154–D158, 2007.

[8] P. Sethupathy, B. Corda, and A. G. Hatzigeorgiou, "TarBase: a comprehensive database of experimentally supported animal microRNA targets," *RNA*, vol. 12, no. 2, pp. 192–197, 2006.

[9] Y. Li, C. Qiu, J. Tu et al., "HMDD v2. 0: a database for experimentally supported human microRNA and disease associations," *Nucleic Acids Research*, vol. 42, no. D1, pp. D1070–D1074, 2013.

[10] X. Chen, M.-X. Liu, and G.-Y. Yan, "RWRMDA: predicting novel human microRNA-disease associations," *Molecular BioSystems*, vol. 8, no. 10, pp. 2792–2798, 2012.

[11] X. Chen, C. C. Yan, X. Zhang, Z. H. You, Y. A. Huang, and G. Y. Yan, "HGIMDA: heterogeneous graph inference for miRNA-disease association prediction," *Oncotarget*, vol. 7, no. 40, pp. 65257–65269, 2016.

[12] J. Xu, C.-X. Li, J.-Y. Lv et al., "Prioritizing candidate disease miRNAs by topological features in the miRNA target–

dysregulated network: case study of prostate cancer," *Molecular Cancer Therapeutics*, vol. 10, no. 10, pp. 1857–1866, 2011.

[13] P. Xuan, K. Han, M. Guo et al., "Prediction of microRNAs associated with human diseases based on weighted k most similar neighbors," *PLoS One*, vol. 8, no. 8, article e70204, 2013.

[14] X. Chen and G.-Y. Yan, "Semi-supervised learning for potential human microRNA-disease associations inference," *Scientific Reports*, vol. 4, p. 5501, 2014.

[15] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, *struc2vec: learning node representations from structural identity*, 2017.

[16] H. Wang, J. Wang, J. Wang et al., "GraphGAN: graph representation learning with generative adversarial nets," *IEEE Transactions on Knowledge and Data Engineering*, 2017.

[17] D. Wang, J. Wang, M. Lu, F. Song, and Q. Cui, "Inferring the human microRNA functional similarity and functional network based on microRNA-associated diseases," *Bioinformatics*, vol. 26, no. 13, pp. 1644–1650, 2010.

[18] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," in *WWW '15: Proceedings of the 24th International Conference on World Wide Web*, Florence Italy, 2015.

[19] A. Grover and J. Leskovec, *node2vec: scalable feature learning for networks*, 2016.

[20] B. Perozzi, R. Al-Rfou, and S. Skiena, *DeepWalk: online learning of social representations*, 2014.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR 2013, International Conference on Learning Representations*, Scottsdale, AZ, USA, 2013.

[22] Ananthram Swami, Nitesh V. Chawla, and Yuxiao Dong, "metapath2vec: scalable representation learning for heterogeneous networks," in *KDD '17: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax NS Canada, 2017.

[23] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[24] T. Mikolov, "Distributed representations of words and phrases and their\n compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.

[25] H. Yin, L. Zou, Q. V. Nguyen, Z. Huang, and X. Zhou, "Joint event-partner recommendation in event-based social networks," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, Paris, France, 2018.

[26] H. Wang, J. Cao, L. Shu, and D. Rafiei, "Locality sensitive hashing revisited: filling the gap between theory and algorithm analysis," in *CIKM'13: 22nd ACM International Conference on Information and Knowledge Management*, San Francisco CA USA, 2013.

[27] L. Bottou, *Stochastic gradient tricks*, 2012.

[28] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 23, no. 23, 2002.

[29] Shaosheng Cao, Wei Lu, and Qiongkai Xu, "GraRep: learning graph representations with global structural information," in *CIKM'15: 24th ACM International Conference on Information and Knowledge Management*, Melbourne Australia, 2015.

[30] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[31] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.