Student Performance Prediction Based on Multi-View Network Embedding

Jianian Li[2], Yanwei Yu[1,*], Yunhong Lu[2], and Peng Song[2]

[1] Ocean University of China, Qingdao, Shandong 266100, China
yuyanwei@ouc.edu.cn
[2] Yantai University, Yantai, Shandong 264005, China

Abstract. Predicting student performance is a very important but yet challenging task in education. In this paper, we propose a Multi-View Network Embedding (MVNE) method for student performance prediction, which effectively fuses multiple data sources. We first construct three networks to model three different types of data sources correlated with student performance, ranging from class performance data, historical grades, to students' campus social relationships. Then we use joint network embedding to learn the embedding representation of students and questions based on the proposed separated random walk sampling. Student performance is predicted based on both student and question similarities in the low-dimensional representation. Experimental results on the real-world datasets demonstrate the effectiveness of the proposed method.

Keywords: Student performance prediction, network embedding, heterogeneous networks, multi-source data.

1 Introduction

Education is the foundation of a nation. Students' performance plays a significant role in a country's social and economic growth by producing creative graduates, innovators and entrepreneurs. In recent years, the phenomenon of failing examinations in universities has become more serious, which has already affected the students' enthusiasm for learning, even the smooth graduation for some students. Predicting students' academic performance in advance has become more important to both students and teachers. On the one hand, educators can strengthen the management of students who do not perform well in predicted results, improving the enthusiasm for the students to learn, and thus reducing the probability of students hanging out. On the other hand, according to the predicted student performance, teachers can adjust the teaching plan in time and facilitate personalized education to enhance the learning efficiency and effect of all students.

Therefore, there are varieties of studies have been conducted to predict student performance. A line of methods use traditional machine learning methods to predict student performance, such as decision tree [17], linear regression [13, 12], Bayesian classification [14], neural network [21], and SVM [1]. Another line of studies use matrix decomposition [18] and collaborative filtering [19] for student performance prediction. The major challenge of performance prediction is to reveal important factors that affect students' academic performance. Several methods [23, 5] are conducted to explore the impact of varieties of potential information on students' academic performance. It has been demonstrated that programming behavior [4], friend relationship [23], and personal behavior [22] are correlated with students' academic performance. There are several studies focus on exploring the effects of multiple variables on student performance, such as the impact of multi-regression model [10] and multi-relational factorization model [18] on student performance prediction. Most of these approaches are supervised learning or semi-supervised learning, which requires plenty of labelled data. Additionally, existing methods only consider a single data source or independently consider the impact of each data source on student performance prediction. But fortunately, thanks to the progress of modern network and information technology, lots of data in the process of teaching and learning has been recorded and collected, such as learning management system data [7], campus behaviors [22] and programming behaviors [4]. Recently, emerging network embedding techniques [8] provide a way to learn features from networks automatically. The basic idea is to learn the low-dimensional representation of nodes in a network by preserving the network structure. There have been many studies on how to embed nodes into a low-dimensional space, such as random walk based methods [15, 11, 25, 24], matrix factorization based methods [2, 16], Random projection based methods [6, 26], and deep learning based methods [20, 3]. However, none of them can handle multi-view network data from multiple different data sources.

In this paper, we propose a multi-view network embedding method for student performance prediction, which supports to predict student performance using multiple data sources. More specifically, we consider students' class practice test records, historical grade data, and campus social relationships as input data. First, we construct a heterogeneous network and two homogeneous networks to model the relationships between students, questions, and students and questions from the three types of data sources. Second, we design a separated random walk sampling for the heterogeneous network, and use joint network embedding to learn the low-dimensional representation of students and questions. Third, we implement a similarity-based performance prediction to estimate students' academic performance using student similarity and question similarity in the low-dimensional representation. Finally, experiments on the real-world datasets demonstrated the effectiveness of our proposed method.

2 Problem Definition

In this section, we first introduce the data used in the paper and then formulate the problem of student performance prediction.

Online Judge (OJ) system is an online test and evaluation platform, which compiles and executes the source code (e.g., C, C++) submitted by users, and verifies the correctness of the program source code through the pre-designed test data. It collects all records of practice tests for all students in the programming courses. We define a practice test record as follows:

Definition 1 (practice test record)**.** *A practice test record is a three tuple* $\langle s_i, q_j, t\rangle$ *that represents student* s_i *taking the time* t *to finish the exercise question* q_j *in OJ system.*

The practice test records reflect the performance of students in class. In addition, the historical grade information of students is also easily available, and thus is often used to predict student performance. We next define a historical grade record as:

Definition 2 (Historical grade record)**.** *A historical grade record is a three tuple* $\langle s_i, c_j, g\rangle$ *that represent student* s_i *achieving the score* g *when he/she took the course* c_j*.*

Moreover, the campus social relationships also influence students' learning activities, which in turn may affect students' performance. We define the campus social relationship network of students as follows:

Definition 3 (Campus Social Relationship Network)**.** *The campus social network is defined as an undirected graph* $\mathcal{G}_s = (V, E_s)$*, where* V *is the set of students, and* E_s *is the set of edges between the students. Each edge* $e_{ij} \in E_s$ *represents the relationship between students* s_i *and* s_j *and is associated with weight* $w_{ij} > 0$*, which indicates their interaction behavior in learning activities (i.e., the interaction frequency).*

In fact, in college life, there are many student activities that may affect students' performance, such as co-participating academic competition, co-completing study topics, co-involving in club activities, and often co-attending self-studies.

One simple but intuitive measure for relationship strength between students is the interaction frequency. Specifically, let $A = \{a_1, a_2, \dots\}$ denote the set of all activities that both students s_i and s_j participate in. Then the weight w_{ij} of edge e_{ij} in graph $\mathcal{G}_s$ is the cardinality of A.

Then we formally define our student performance prediction problem as follows:

Problem (Student Performance Prediction Problem)**.** *At semester* Γ*, given the set of practice test records of all student in the semester, the historical grade records , and the campus social network* $\mathcal{G}_s$*, our goal is to predict the students' academic performance rank at this semester.*

3 Method

3.1 Overview

Figure 1 shows the overall framework of our proposed model based on multi-view network embedding, which includes three major modules as follows: *Multi-view* *heterogeneous network construction* module builds a heterogeneous network that involves students and tested questions to capture all relationships among students, questions, and student and questions from input data. *Network embedding* module learns the low-dimensional representations for students and questions using our proposed separated random walk sampling from the constructed heterogeneous network. *Similarity-based performance predictor* module finally predicts students' academic performance based on both student similarity and question similarity in the low-dimensional vector space.

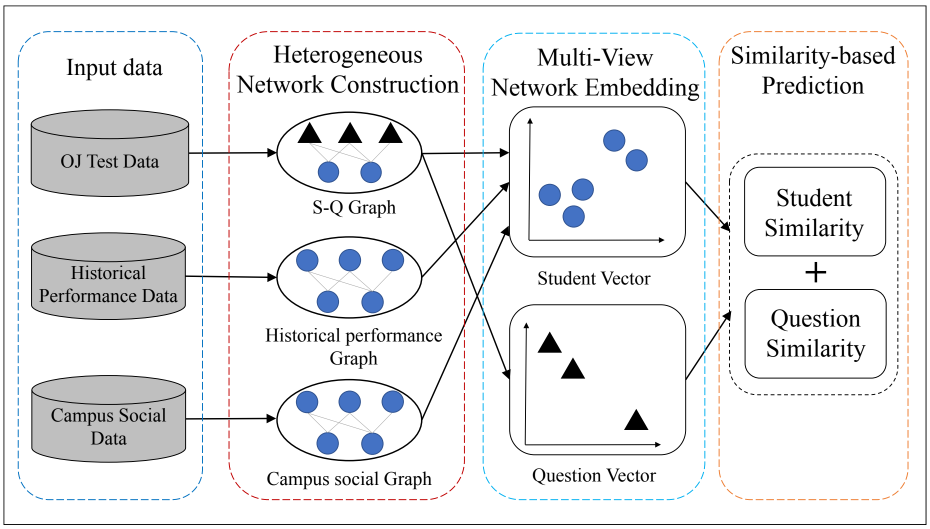


Fig. 1: Overview of the proposed multi-view embedding based student performance prediction model.

3.2 Heterogeneous Network Construction

We first introduce our student-question (or in short *S-Q*) graph as follows:

Definition 4 (Student-Question Graph). *Student-Question (S-Q) graph is a bipartite graph $\mathcal{G}_{sq} = (S \cup Q, E_{sq})$, where S is the set of students and Q is the set of tested questions. E_{sq} is the set of edges between students and questions. If student s_i did tested question q_j, there will be an edges e_{ij} between them, otherwise none. The weight w_{ij} on edge e_{ij} is set to t according to each practice test record $\langle s_i, q_j, t\rangle$.*

The *S-Q* graph is designed to capture all information of students' class performance in this semester.

Next we also build a historical information network to capture the student similarities from student historical performance data. Specifically, the historical information network is an undirected graph $\mathcal{G}_h = (S, E_h)$, where S is the set of all students, and E_h is the set of edges between students. If two students achieve relative equivalent scores (e.g., less than 5 percentage points) in the same course, we regard they are similar once. The weight w_{ij} on edge $eij \in E_h$ is the sum of similarities between students s_i and s_j performed in the historical grade records.

The three types of graphs above (i.e, S-Q graph, historical graph, and campus social graph) can well capture the class performance influence, historical performance effect and campus social relationship effect, respectively. Therefore, we propose to learn embeddings from the three graphs jointly to estimate the student similarity and question similarity.

3.3 Multi-view Network Embedding

The three graphs collaboratively model the similarity relationships between students from multiple perspectives. Inspired by the recent graph embedding techniques [15, 11], we propose a multi-view network embedding method based on random walk sampling.

Actually, our constructed heterogeneous graph is not a pure similarity graph. S-Q graph only captures the information that students taking how long time to finish the tested questions, which does not represent the similarity relationships between students and questions, and thus can not indicate the similarity between students. Therefore, existing random walk sampling strategies are not applied into our graph directly.

To capture the similarities among students and among questions in S-Q graph, we design a separated walk sampling strategy which guides the generation of random walks in S-Q graph: Consider a random walk that just traversed edge (s_1,q), and now stays at node q (Figure 2). The walk now needs to decide on the next node so it evaluates the transition probabilities $Pr(q, s_i)$ on edges (q,s_i) starting from q. We set the transition probability $Pr(q, s_i)$ as follows:

$$Pr(q, s_i) = \begin{cases} 1 & s_i = s_1 \\ \exp(\lambda \frac{\min(w(s_1,q),w(q,s_i))}{\max(w(s_1,q),w(q,s_i))}) & |w(s_1, q) - w(q, s_i)| < r \\ \exp(-\frac{\max(w(s_1,q),w(q,s_i))}{\min(w(s_1,q),w(q,s_i))}) & otherwise \end{cases} \tag{1}$$

Intuitively, parameters r and λ flexibly control how to explore the student neighbors of node q. In particular, r controls the walk to tend to visit the student nodes who take similar time with s_1 w.t.r. question q. λ allows the search to differentiate between similar nodes and dissimilar nodes by scaling the transition probability. But even if there no similar student node in the next step, the return weight (i.e.,1) also ensures the walk to backtrack a step, rather than going to the dissimilar student nodes.

Note that, $Pr(q, s_i)$ is not real transition probability in the strict sense, but an updated transition weight, hence the transition probability will be computed based on the updated weights.

By sampling the random walks in S-Q graph, we can collect the walks in which student node and question node alternate. However, the sampled walks are not suitable for being used directly to learn the embedding vectors of students and questions. First, it is not necessary to map students and questions into the

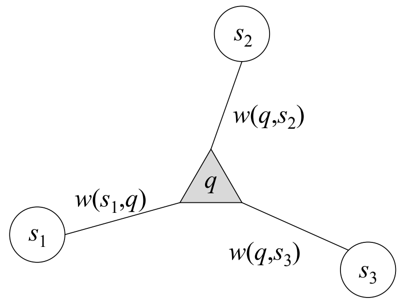


Fig. 2: An example of walk sampling in S-Q graph.

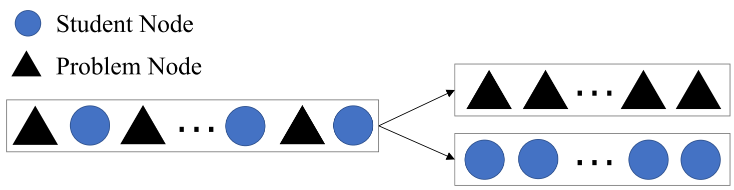


Fig. 3: Illustration of the separated random walk sampling in S-Q graph.

same vector space. That is, we need not learn the similarity between students and questions (i.e., the first-order proximity in the walks). Second, directly applying such walks actually hurts the prediction accuracy, because such walks invisibly closes the representations between dissimilar student nodes. For example, two completely dissimilar students may have done the same questions. Because they take relatively large time differences, they may not appear in the same walks. However, in different walks, both student nodes may have first-order proximity with the same question, and thus their representations tend to be similar during the training process.

To address these issues, we propose a separated random walk sampling strategy. As shown in Figure 3, we generate two walks from an original walks sampled from S-Q graph, namely, each walk only contains one kind of nodes. In this way, we cut off the similarity between students and questions, and thus effectively differentiates the vector representations between dissimilar student nodes, and between dissimilar question nodes.

To obtain the multi-view embedding representations, we again generate random walks in historical information graph $\mathcal{G}_h$ and campus social graph $\mathcal{G}_s$, and then jointly learn the embeddings of student nodes and question nodes using the sampled random walks from the three graphs.

Actually, we can easily tune the weights for the three views in the embedding learning by setting different number of sampled walks on three graphs in the implementation. More specifically, we fix the number of sampled walks on historical graph $\mathcal{G}_h$, and set α and β times the number of walk samples for S-Q graph and $\mathcal{G}_s$, respectively. By default, we set $\alpha = \beta = 1$.

3.4 Similarity-based Performance Prediction

After embedding different types of information into the representation, we use the cosine distance to measure the similarity in the embedding space.

To predict the performance rank of students, we predict the total time took by students to finish random selected questions, and use the time to get students' performance ranking.

Given a selected question q, the time spent $t_i(q)$ by student s_i is predicted by:

$$t_i(q) = \omega \frac{\sum\limits_{s_j \in N_k(s_i)} t_j(q)}{k} + (1-\omega) \frac{\sum\limits_{q_j \in N_k(q)} t_i(q_j)}{k}, \tag{2}$$

where $N_k(s_i)$ represents the kNNs of s_i in the embedding space, and $N_k(q)$ denotes the most similar k questions of q. By default, ω is set to 0.5.

4 Experiment

4.1 Datasets

We collected the data of nearly two thousand students from one university in China during 2016/09/01 to 2018/07/15. The dataset consists of three types of data, which are described as follows:

- OJ practice test data: This data contains almost 2.1 millions test records from more than 5,000 students.
- Historical performance data: This data contains all historical course grades for the selected students in their first two years of college. Specifically, the data includes the grade information of 24 subjects, such as advanced mathematics, analog circuits, and linear algebra.
- Campus social data: We collect 25 campus activities of the selected students reflecting the campus social relationships among students, such as subject competitions, lab-mates, learning group.

We divide the data into two datasets, one containing the above three types of data, denoted Data A. It includes more than 300 students from two majors. Another dataset includes only the first two types of data, denoted Data B, which consists of more than 1400 students from two grades.

4.2 Baselines and Metrics

We compare our method with the following baselines:

- Average-based methods: This baseline uses average time spent to estimate the performance of students. We compare two average-based methods: one is using the average time spent per student on all his/her answered questions, the other one is using the average time spent of each student on the questions that are similar with examinations to predict ranking, which are referred to as Global-Avg and Neighbor-Avg, respectively.

– Matrix Factorization (MF) [19]: We perform MF on student-question matrix instead of the user-item matrix, and complete the student-question matrix by matrix decomposition.
– Collaborative Filtering (CF) based methods [19]: This baseline contains two methods: User-CF and Item-CF.

Our embedding method has three kinds of variations:

– Single-view variations: We have three single-view variations, that is, OJ-view, History-view, and Social-view.
– Dual-view variations: We also have three dual-view variations, each variation considers both two different data sources.
– MVNE/s: MVNE/s directly use the original random walk sampling in MVNE, rather than the separated random walks.

Finally, we predict the results of all students from two majors and use RMSE (root mean square error) to evaluate the effectiveness of our proposed method, which was also used as an assessment metric in previous work [5, 18, 19, 9].

Table 1: Experimental results of all methods

Method	Data A		Data B	
	Major I	Major II	Grade I	Grade II
Global-Avg	0.3219	0.3184	0.3795	0.3651
Neighbor-Avg	0.3109	0.3073	0.3755	0.3342
User-CF	0.3023	0.3287	0.3545	0.3604
Item-CF	0.2945	0.2743	0.3504	0.3434
MF	0.2901	0.2895	0.3277	0.3012
Social-view	0.2962	0.2740	–	–
History-view	0.2619	0.2431	0.2586	0.2565
OJ-view	0.2195	0.2173	0.2456	0.2286
Social-History-view	0.2342	0.2201	–	–
Social-OJ-view	0.1934	0.1945	–	–
OJ-History-view	0.1756	0.1843	0.2258	0.2056
MVNE/s	0.2975	0.2886	0.3456	0.3256
MVNE	0.1691	0.1703	–	–

4.3 Experimental Results

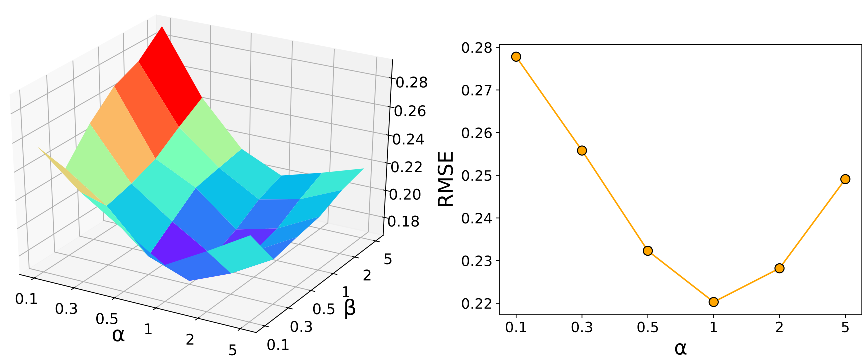
Overall Performance Table 1 shows the experimental results of our method compared to all baselines.

As we can see, our proposed Multi-View Network Embedding (MVNE) method significantly outperforms all baselines on two datasets. The main reasons may be: First, MVNE effectively combines three different types of data sources (i.e., historical performance data, campus social data, and class performance records), while all existing methods consider only a single data source. Second, MVNE uses a heterogeneous networks to clearly reflect the relationships between students and questions in the real-world datasets, and uses joint network embedding to encode all data source information into a low-dimensional representation for each student. Last but most important, MVNE uses the proposed separated random walk sampling in S-Q graph to significantly improve the embedding representations of students and questions in predicting student performance.

Variation Study From Table 1, we also observe that OJ-view performs better than other two single-view methods (i.e., Social-view and History-view). This may be because the class performance has greater impact on students' final performance in this semester compared to historical grade and campus relationship. Among the three views, campus social relationship has the weakest impact on student performance. However, Social-view method using network embedding is still better than other baselines.

Moreover, dual-view methods perform better than corresponding either of single-view methods on both two datasets. And MVNE achieves the best performance on both two majors. This also demonstrates that our proposed multi-view embedding method maximally fuses the information from multiple data sources.

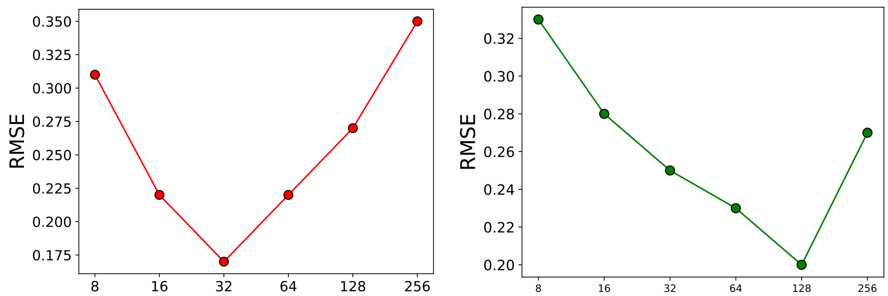
Additionally, MVNE performs significantly better than MVNE/s on two majors, which also confirms that our proposed separated random walk sampling is much more useful in learning the representations of students and questions.



(a) Data A (b) Data B

Fig. 4: Results of varying the weight parameters

Parameter sensitivity First, we study the impact of each view on the prediction performance by varying the weights of the three views in the embedding



(a) Data A (b) Data B

Fig. 5: Performance w.r.t. the embedding dimension

learning. The results of parameters α and β on two datasets are shown in Figure 4. We vary α and β from 0.1 to 5 respectively. We use the grid search method to find the best parameter settings. As we can see, the RMSE of MVNE first decreases to the minimal value and then increases as the weight parameters increasing. This is intuitive because both class performance and social relationships are essential for a precise prediction. As shown in Figure 4(a), the RMSE reaches minimum value when α and β fall around 1 and 0.3, respectively.

Similarly, the RMSE achieves minimum value when α is 1 in Figure 4(b). In addition, it is clear that the prediction error decreases rapidly with α increasing from 0. This suggests the class performance contributes a lot to the overall prediction accuracy.

Second, we explore how the performance of MVNE changes with respect to embedding dimension. The results on two datasets are shown in Figure 5. As expected, the performance of MVNE first increases as embedding dimension increases, and then drops when the dimension becomes too large. We also observe that our MVNE achieves the best results when selecting 32 dimensions on the small dataset (Figure 5(a)), and the best performance when adopting 128 dimensions on the large dataset (Figure 5(b)).

5 Conclusions

Predicting student performance is a very important and challenging task in education. This paper proposes a multi-view based network embedding method for student performance prediction. Specifically, we use joint network embedding to learn the similarities between students, questions, and students and questions from three different types of data sources. We also design a separated random walk sampling in the heterogeneous graph to improve the prediction performance. Finally, the superiority of the proposed model is confirmed by experiments on the real-world datasets.

Acknowledgments. This work is partially supported by the National Natural Science Foundation of China under grant Nos. 61773331, 61403328 and 61703360.

The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

1. Al-Shehri, H., Al-Qarni, A., Al-Saati, L., Batoaq, A., Badukhen, H., Alrashed, S., Alhiyafi, J., Olatunji, S.O.: Student performance prediction using support vector machine and k-nearest neighbor. In: 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE). pp. 1–4. IEEE (2017)
2. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM international on conference on information and knowledge management. pp. 891–900 (2015)
3. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: Thirtieth AAAI conference on artificial intelligence (2016)
4. Carter, A.S., Hundhausen, C.D., Adesope, O.: The normalized programming state model: Predicting student performance in computing courses based on programming behavior. In: Proceedings of the eleventh annual International Conference on International Computing Education Research. pp. 141–150 (2015)
5. Cetintas, S., Si, L., Xin, Y.P., Tzur, R.: Probabilistic latent class models for predicting student performance. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 1513–1516 (2013)
6. Chen, H., Sultan, S.F., Tian, Y., Chen, M., Skiena, S.: Fast and accurate network embeddings via very sparse random projection. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 399–408 (2019)
7. Conijn, R., Snijders, C., Kleingeld, A., Matzat, U.: Predicting student performance from lms data: A comparison of 17 blended courses using moodle lms. IEEE Transactions on Learning Technologies **10**(1), 17–29 (2017)
8. Cui, P., Wang, X., Pei, J., Zhu, W.: A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering **31**(5), 833–852 (2018)
9. Elbadrawy, A., Polyzou, A., Ren, Z., Sweeney, M., Karypis, G., Rangwala, H.: Predicting student performance using personalized analytics. Computer **49**(4), 61–69 (2016)
10. Elbadrawy, A., Studham, R.S., Karypis, G.: Collaborative multi-regression models for predicting students' performance in course activities. In: proceedings of the fifth international conference on learning analytics and knowledge. pp. 103–107 (2015)
11. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864. ACM (2016)
12. Huang, S., Fang, N.: Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models. Computers & Education **61**, 133–145 (2013)
13. Kabakchieva, D.: Predicting student performance by using data mining methods for classification. Cybernetics and information technologies **13**(1), 61–72 (2013)
14. Mayilvaganan, M., Kalpanadevi, D.: Comparison of classification techniques for predicting the performance of students academic environment. In: 2014 International Conference on Communication and Network Technologies. pp. 113–118. IEEE (2014)

15. 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)
16. Qiu, J., Dong, Y., Ma, H., Li, J., Wang, C., Wang, K., Tang, J.: Netsmf: Large-scale network embedding as sparse matrix factorization. In: The World Wide Web Conference. pp. 1509–1520 (2019)
17. Quadri, M.M., Kalyankar, N.: Drop out feature of student data for academic performance using decision tree techniques. Global Journal of Computer Science and Technology (2010)
18. Thai-Nghe, N., Drumond, L., Horváth, T., Schmidt-Thieme, L., et al.: Multi-relational factorization models for predicting student performance. In: KDD Workshop on Knowledge Discovery in Educational Data (KDDinED). pp. 27–40 (2011)
19. Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., Schmidt-Thieme, L.: Recommender system for predicting student performance. Procedia Computer Science **1**(2), 2811–2819 (2010)
20. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1225–1234 (2016)
21. Wang, T., Mitrovic, A.: Using neural networks to predict student's performance. In: Proceedings of International Conference on Computers in Education. pp. 969–973. IEEE (2002)
22. Yao, H., Lian, D., Cao, Y., Wu, Y., Zhou, T.: Predicting academic performance for college students: A campus behavior perspective. arXiv preprint arXiv:1903.06726 (2019)
23. Yao, H., Nie, M., Su, H., Xia, H., Lian, D.: Predicting academic performance via semi-supervised learning with constructed campus social network. In: Proceedings of International Conference on Database Systems for Advanced Applications. pp. 597–609. Springer (2017)
24. Yu, Y., Wang, H., Li, Z.: Inferring mobility relationship via graph embedding. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **2**(3), 1–21 (2018)
25. Yu, Y., Yao, H., Wang, H., Tang, X., Li, Z.: Representation learning for large-scale dynamic networks. In: International Conference on Database Systems for Advanced Applications. pp. 526–541. Springer (2018)
26. Zhang, Z., Cui, P., Li, H., Wang, X., Zhu, W.: Billion-scale network embedding with iterative random projection. In: 2018 IEEE International Conference on Data Mining (ICDM). pp. 787–796. IEEE (2018)