Knowledge-Based Systems 281 (2023) 111071

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Personas-based Student Grouping using reinforcement learning and linear programming

Shaojie Ma [a], Yawei Luo [a,*,1], Yi Yang [b]

[a] School of Software Technology, Zhejiang University, China
[b] College of Computer Science and Technology, Zhejiang University, China

ARTICLE INFO

Keywords:
Student grouping
Personas-based grouping
Reinforcement learning
Linear programming
Collaborative learning

ABSTRACT

Group discussions and assignments play a pivotal role in the classroom and online study. Existing research has mainly focused on exploring the educational impact of group learning, while the study on automated grouping still remains under-explored. This paper proposes a principled method that aims to achieve personalized, accurate, and efficient grouping outcomes. Dubbed as **P**ersonas-based **S**tudent **G**rouping (PSG), our method first applies unsupervised clustering techniques to assign personas to students based on their behavioral characteristics. Based on their personas, we then utilize deep reinforcement learning to search for appropriate grouping rules and perform linear programming to obtain a suitable grouping scheme. Finally, the teaching effectiveness is fed back as the rewards to the reinforcement learning model to optimize future grouping scheme selections. Extensive experiments conducted on MOOCs datasets show that PSG can achieve more advantageous performance in both efficiency and effectiveness compared to the manual or random grouping mechanism. We hope PSG can provide students with a more enhanced learning experience and contribute to the future development of education. Our project homepage is available at https://PSG-project.pages.dev.

1. Introduction

Group discussions and assignments are indispensable in the modern classroom and online study paradigms. Proper grouping can effectively facilitate collaborative learning, uncover students' potential and enhance their teamwork abilities, thus promoting their all-round development [1–3]. In traditional classroom scenarios, experienced teachers often rely on their familiarity with students' personalities, strengths, and weaknesses to manually compose groups [4]. However, teachers can hardly recognize every student's characteristics and requirements accurately in large classrooms or remote learning environments, such as Massive Open Online Courses (MOOCs), making manual personalized grouping at scale impractical. This limitation has sparked research into using advanced algorithms and big student data to automate grouping.

Over the past few decades, research on computer-assisted instruction has emerged with the advancement of computer technology. One such area of investigation is Computer-Supported Collaborative Learning (CSCL), which encompasses using technology to analyze, comprehend, and enhance collaborative learning processes. CSCL research focuses on developing algorithms to automatically form effective student groups based on available data, enabling data-driven personalized grouping [5]. However, CSCL primarily focuses on incorporating computer-assisted steps into collaborative teaching, serving only as a part of human instruction and providing supplementary assistance [6–8]. As a result, these methods heavily rely on human control and cannot automatically improve based on teaching outcomes once detached from human intervention. These limitations have restricted the further advancement of research related to CSCL.

In this paper, we aim to design an intelligent method that facilitates large-scale and personalized grouping. Different from vanilla CSCL studies, we propose a Personas-based Student Grouping (PSG) method, which harnesses the strengths of deep reinforcement learning (DRL) and linear programming (LP) to devise a comprehensive, fully automated process that can operate without human intervention. Specifically, PSG first applies unsupervised clustering techniques to assign personas to students based on their behavioral characteristics. Next, we utilize reinforcement learning techniques to select appropriate grouping rules and perform linear programming to obtain a suitable grouping scheme. Finally, the teaching effectiveness can be fed back to the reinforcement learning model to optimize future grouping scheme selections.

PSG is particularly well-suited for teaching scenarios with many students and a few teachers, such as in MOOCs. It greatly simplifies

* Corresponding author.
E-mail address: yaweiluo@zju.edu.cn (Y. Luo).
[1] This work is supported by National Key R&D Program of China under Grant No. 2020AAA0108800.

https://doi.org/10.1016/j.knosys.2023.111071
Received 29 July 2023; Received in revised form 26 September 2023; Accepted 12 October 2023
Available online 14 October 2023

teachers' workload and only requires specific rules based on their expertise at the outset of the algorithm's operation. After teaching, teachers can also evaluate the teaching effectiveness and obtain insight from the grouping results for self-improvement. The internal workings of the algorithm do not depend on human involvement, resulting in improved efficiency and scalability of the grouping process. Our contributions can be summarized as follows:

- Personas-based Clustering: By clustering students in terms of personas, PSG projects the high-dimensional behavioral embeddings into personas prototypes, which greatly simplifies the state space of deep reinforcement learning. The assigned personas also offer valuable insight to teachers on further manual grouping scenarios.
- Integration of Reinforcement Learning and Linear Programming: The linear programming component computes optimal grouping solutions based on certain predefined rules, while the reinforcement learning component learns from linear programming results and teaching feedback to generate gradually better grouping rules, leading to more enhanced teaching outcomes. To our best knowledge, we are the first to harness the strengths of deep reinforcement learning and linear programming in student grouping tasks.
- Efficient and Effective Student Grouping: Extensive experiments conducted on MOOCs datasets show that PSG can achieve more advantageous performance in both efficiency and effectiveness compared to the manual or random grouping mechanism.

2. Related work

2.1. Collaborative learning

Collaborative learning offers an effective way to improve student engagement and study outcomes. Researchers have reviewed the effectiveness of cooperative learning in higher education, studied the impact of different goals on adolescents' achievement and relationships, and explored the influence of partnerships on programming teaching effectiveness [1–3]. In a general collaborative learning scenario, teachers may group students based on various attributes, including academic performance, gender, personality traits, and friendship preferences. Common grouping approaches include ability-based grouping, mixed ability grouping, interest-based grouping, and random assignment [4,9,10]. Ability grouping involves grouping students by perceived academic capabilities, often using test scores or grades Chiu et al. [11]. Mixed ability grouping combines students with diverse skill levels and is often favored for promoting peer learning and modeling Murphy et al. [12]. Additional work has studied the impact of homogeneous versus heterogeneous grouping on collaborative learning outcomes [9,13,14].

Previous research has also employed various techniques to facilitate student grouping and performance prediction in Computer-Supported Collaborative Learning environments. Spoelstra et al. [15] and Moreno et al. [16] used questionnaires to gather student characteristics and employed genetic algorithms for team formation. Cen et al. [17] employed collaborative learning platform log data and classification and regression algorithms to forecast group performance. These diverse approaches have demonstrated the effectiveness of leveraging various data sources and algorithms to enhance CSCL interactions and performance prediction.

Recent research has explored how different grouping schemes can optimize collaborative learning. Li and Shan [18] proposed an algorithm considering communication capacities and social networks, enabling user feedback to improve groups iteratively. However, this was tailored to something other than classroom environments. Haq et al. [19] demonstrated the potential of knowledge-based dynamic grouping to improve collaborative learning by forming initial groups based on assessed learning styles and knowledge, then dynamically rebalancing groups based on relative knowledge levels. Zheng et al. [7] designed an integrated mathematical model and an improved genetic algorithm to solve the model and obtain optimal learning groups to meet various grouping requirements for different educational contexts. In the research of Xu et al. [8], different grouping schemes were investigated to understand their influences on promoting active learning and enhancing student programming skills through peer learning. Their research highlighted the positive impact of thoughtfully designed grouping schemes compared to other grouping methods. However, they still need to present a complete automated grouping scheme since the grouping process relies on manual operations. Reinforcement learning (RL) provides a potential solution where agents learn optimal grouping behaviors through ongoing environment interactions [6].

In PSG, we follow the insight of [6] to employ RL as our student grouping mechanism generator. However, PSG goes beyond it by combining reinforcement learning and a simplex algorithm instead of a genetic algorithm to achieve better performance.

2.2. Reinforcement learning

Reinforcement learning (RL) has demonstrated remarkable achievements in educational contexts, ranging from intelligent tutoring systems to personalized learning platforms. Notably, Chi et al. [20] conducted empirical research evaluating the application of RL to induce effective and adaptive pedagogical strategies. Similarly, Iglesias et al. [21] contributed to the field by developing RL-based pedagogical policies for adaptive and intelligent educational systems. Additionally, through their modular RL framework, Rowe and Lester [22] made significant strides in enhancing student problem-solving within narrative-centered learning environments. These studies collectively underscore the potential of RL as a valuable tool for revolutionizing educational practices and promoting personalized, effective learning experiences.

For grouping problems precisely, reinforcement learning has shown promise for adapting group configurations over time. Panait and Luke [23] used reinforcement learning for cooperative multi-robot learning, showing how it enables robots to learn varying group sizes and specialization of roles. Bassen et al. [24] demonstrates an RL model to schedule real-time educational activities for an extensive online course through active learning. Omidshafiei et al. [25] used decentralized multi-agent RL to learn complex grouping behaviors.

These studies above demonstrate reinforcement learning's capabilities for automated grouping tasks. However, RL for online grouping still needs to be further explored, especially for evolving classroom conditions. Our proposed approach aims to leverage the benefits of RL for education grouping.

2.3. Clustering algorithms

Clustering algorithms have become essential for exploratory data analysis across many domains. These unsupervised methods aim to organize unlabeled data into meaningful groups or clusters based on similarity to uncover hidden structures in the data, from Xu and Wunsch [26]. Clustering has a long history spanning computer science, statistics, and machine learning, with linkage to fields like biology, psychology, and economics [27–29].

Plenty of clustering-based machine learning algorithms have been proposed in history, wherein K-means is one of the most popular and widely used clustering algorithms due to its simplicity and efficiency [27]. This prototypical partitioning technique minimizes the squared error objective between points and their cluster centroids [30]. Limitations of K-means include sensitivity to initialization, convergence to local optima, and bias toward spherical clusters of similar size [31]. Recent advances focus on seeding strategies, handling constraints, theoretical analysis, and scalability via sampling and parallelization [32]. X-means is an extended version of K-means that estimates the number of clusters K based on Bayesian Information Criterion (BIC) [33]. The

more recent work such as CAS, supports inheritance and exceptions without probabilistic assumptions [34], and GBS, a graph-based system for multi-view clustering [35]. Additionally, new dissimilarity measures have been defined, including one based on biological taxonomy and rough set theory [36], and attribute selection methods like MDA, considering attribute dependency [37]. Moreover, heuristic algorithms have been introduced, incorporating prior knowledge of cluster size [38], and fuzzy k-prototypes algorithms have been designed to handle mixed numeric and categorical data [39].

In a more specific domain, e.g. education, as our focus in this paper, clustering method helps analyze patterns in student data to improve learning and institutional effectiveness [40]. For example, clustering student usage logs from learning management systems can inform personalized e-learning tools [41]. In intelligent tutoring systems, clustering student problem-solving steps aids knowledge tracing [42]. Clustering also enables analyzing social networks, collaboration styles, and discussion forum patterns, [43].

Overall, clustering research continues to expand into new disciplines with novel data types and goals. Key directions include scalability, robustness, emerging data sources like text and social networks, integrative clustering of varied data, and interpretability for real-world impact.

3. Methodology

3.1. Overview

PSG adopts a two-step structure, as illustrated in Fig. 1. In Step 1, we collect students' multidimensional behavioral information from the MOOCs platform as input data. After processing this information, personas labels like "leader" or "collaborator" are assigned for different clusters.

In Step 2, we propose to combine reinforcement learning and linear programming for grouping. Initially, we design a DQN model where the *state* (observation) contains various grouping rules and the *action* (output) represents the chosen rule. These rules are generated based on the personas labels obtained in Step 1. Once we have a good subset of rules for selecting groupings, we perform linear programming with these rules. The output of the linear programming gives us the final grouping scheme, which is then implemented in classroom teaching. During classroom teaching, we observe the teaching outcomes of different grouping rules (i.e. different types of groups). This information is utilized to provide feedback as the *reward* to the reinforcement learning model and simultaneously update rule scores. Through multiple iterations, PSG gradually generates grouping rules more suited for effective teaching.

3.2. Generate personas labels

In previous student grouping practices, some experienced teachers have demonstrated proficient skills in manually composing student groups. They first assess students' traits and accordingly assign personas labels like "leader", "executor" or "coordinator". Subsequently, using these personas labels, they group students based on certain rules. For example, forming a group of four with one leader, two executors, and one coordinator. This process is reasonable since it simplifies the complexity of grouping by utilizing simple rules while considering the individual traits of students for a personalized approach. Teachers would summarize some rules from past teaching experience, for instance:

```
1 leader + 2 executors + 1 coordinator = good group

3 leaders + 1 observer = bad group
```

These rules provide valuable and straightforward guidelines for the teacher.

To better leverage teachers' expertise, we referred to this methodology. We adopt a clustering-based approach to assign personas labels to students. With students' personas labels obtained, we can then utilize the grouping experience accumulated by teachers. Specifically, we utilize the k-means algorithm for clustering. We first cluster students based on their behavioral features and then annotate different clusters according to teacher experience. For input features, we select students' historical performance data on the MOOCs platform as relevant features, including scores, in-class performance, participation level, etc. To ensure meaningful clustering, we perform feature engineering on these multidimensional data and utilize the z-score standardization method to standardize each feature dimension.

The choice of cluster number K directly influences the final clustering outcome. To determine the optimal K, we referred to the XMeans method by Pelleg and Moore [33]. XMeans is a k-Means-based automatic clustering algorithm that incrementally splits clusters using BIC criteria to determine the optimal number of clusters. Once the cluster number K is selected, we apply the k-means algorithm for clustering, dividing students into K clusters. The k-means algorithm minimizes total within-cluster variance by iteratively updating cluster centers.

Specifically, we randomly select K samples as initial cluster centers, denoted as $C_1, C_2, \ldots, C_K$. At each iteration, for every data point x in the dataset, we compute its distance to each cluster center and assign it to the cluster with minimum distance:

$$class(x) = \arg\min_i |x - C_i|. \tag{1}$$

The cluster centers are then updated by calculating the mean of samples in each cluster, obtaining new cluster centers:

$$C_i = \frac{1}{N_i} \sum_{x \in Cluster_i} x, \tag{2}$$

where N_i denotes the number of samples in the ith cluster. Thus, the average of all samples in a class becomes the new cluster center. This process is repeated multiple times until the cluster centers no longer change or a pre-defined iteration limit is reached.

After completing the clustering process, the clusters are mapped to personas labels. Based on teachers' expertise, we label the K clusters as personas like "leader", "executor" or "coordinator". Teachers can inspect the clustering results and assess the distribution of features in clusters to determine reasonable personas mappings.

3.3. Deep reinforcement learning loop design

Through clustering, we have assigned each student a personas label, allowing us to leverage some human-summarized prior knowledge. PSG will generate all possible rules, and score them based on the human-summarized rules. However, the number of generated groupings can be huge (reaching A^N scale), making it infeasible to directly use in the subsequent optimizer.

To address the combinatorial explosion of grouping rules, we propose a deep reinforcement learning-based automatic rule selection method. We introduce this reinforcement learning-based algorithm in five parts: state space, action space, deep Q network, reward function, and training process.

Action Space: The action space consists of all candidate rules, which we denote as an linear set $A = 1, 2, \ldots, C$, where each linear represents the index of a corresponding candidate rule. An action $a \in A$ represents selecting to add a new rule on top of the current rule set. That is, the actions represent picking a new additional rule from the pool of candidate rules to include in the current set. Initially, the rule set is empty. Complete rule subsets are formed through the cumulative addition of multiple actions.

State Space: Each state s represents the current chosen subset of rules. The state can be represented as a C-dimensional 0–1 vector,

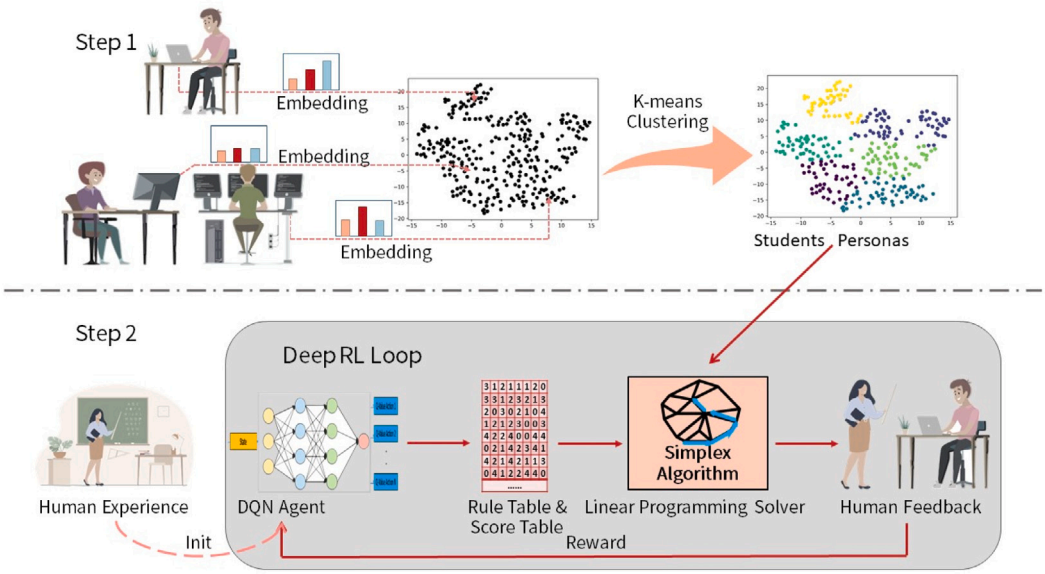

Fig. 1. Flowchart of PSG.

where the ith position being 1 indicates rule i has been selected. The initial state s_0 is an all-0 vector, denoting an empty rule set. After taking an action, the corresponding rule position is updated to 1, indicating its addition to the set.

Reward Function: We designed a polynomial reward function with three terms:

$$r(s,a) = r_{eval} - r_{repeat} - r_{fail}, \tag{3}$$

where r_{eval} represents the score based on evaluating the linear programming grouping outcome using the current rule subset, rewarding better combinations; r_{repeat} is a penalty term for repeatedly selecting the same rule; r_{fail} is a penalty when grouping fails.

Deep Q Network: We construct a multilayer perceptron to approximate the state–action value function $Q(s,a)$. The network takes in a one-step state s as input, and outputs predicted Q values for each action a. It contains two fully-connected layers, each followed by a ReLU activation function. The output layer size equals the action space size C to predict Q values for all possible actions.

The network parameters θ are optimized by regressing to Q targets:

$$L(\theta) = (r + \gamma * \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta))^2, \tag{4}$$

where s, a are current state–action, r is immediate reward, s' is next state, γ is discount factor. Target network parameters θ^- are periodically copied from the current network.

Training Process: The algorithm starts from initial state s_0 and iterates Algorithm 1.

Algorithm 1 Deep Reinforcement Learning for Rule Selection

Require: Candidate rule set R, Maximum number of rules N
Ensure: Selected optimized rule subset S
Initialize replay memory D
Initialize policy network Q, random weights θ
Initialize state s_0 as empty set
for episode $= 1, 2, ..., E$ **do**
  Receive initial state s_0
  for $t = 1, 2, ..., T$ **do**
    With probability ϵ select random action a
    Otherwise select $a = \arg\max_a Q(s, a; \theta)$
    Execute a, add rule r_a to current set S
    Observe reward r and new state s'
    Store transition (s, a, r, s') in D
  end for
  if $|S| == N$ **then**
    break
  end if
  Sample batch randomly from D
  Update θ via batch regression
end for
Output optimized rule subset S
Calculate rewards from teaching feedback, add to D, update θ
Update $Score$

3.4. Generating grouping schemes

The optimized rule subset obtained from the reinforcement learning model consists of a rule matrix and a score matrix. The rule matrix $Rule$ is sized $R \times K$, where R is the number of rules and K is the number of clusters. $Rule_{i,j}$ denotes the number of students of cluster j contained in rule i. The score matrix $Score$ is sized R, where $Rule_i$ represents the score for rule i.

Additionally, from the clustering step we can obtain the student count matrix $Student$ of size K, where $Student_i$ denotes the number of students with personas i. Based on matrices $Rule$ and $Student$, we establish the following linear programming model to solve for the usage count of each rule:

$$\underset{x}{\text{maximize}} \quad \sum_{i=1}^{n} Score_i \cdot x_i, \tag{5a}$$

$$\text{subject to} \quad \sum_{i=1}^{n} Rule_{ji} \cdot x_i = Student_j \quad \text{for } j = 1, 2, \ldots, K, \tag{5b}$$

$$x_i \geq 0 \quad \text{for } i = 1, 2, \ldots, R. \tag{5c}$$

The objective (5) is to maximize the total score based on rule usage counts. Constraint (5b) ensures each student is assigned to groups according to their personas counts. Constraint (5c) enforces non-negativity of the rule usage counts. To solve this linear programming

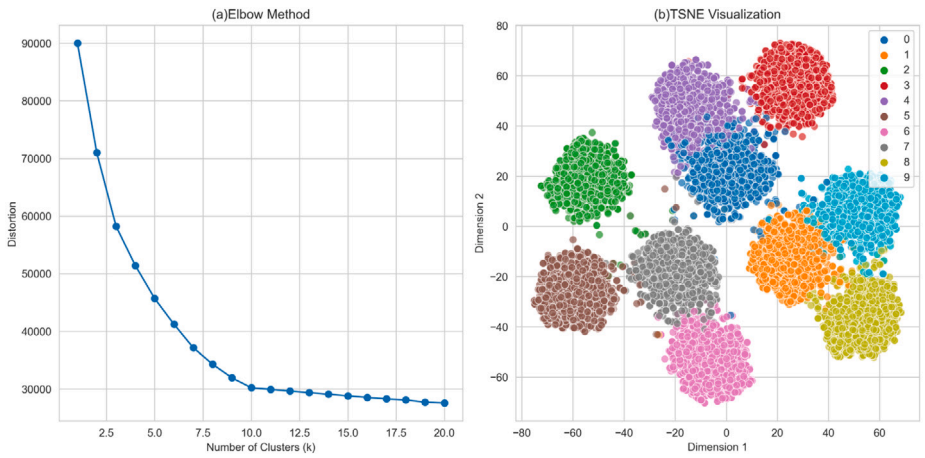

Fig. 2. Elbow and t-SNE visualizations.

model, we utilize the PuLP library in Python, a fast implementation of simplex algorithm [44].

Solving this optimized configuration of rule usage counts yields a high-quality grouping scheme. The scalability of clustering and flexibility of DRL are combined successfully to improve grouping efficiency, enabling PSG to maintain performance in large-scale student grouping scenarios.

4. Experiments

4.1. Experimental setup

Dataset: To comprehensively evaluate PSG, we generated a dataset of 10,000 students referring to a MOOCs platform's student feature distribution and conducted experiments on it. Each student is represented by a 10-dimensional feature vector. These features are standardized to z-scores. The experimental goal is to divide these students into groups of equal size. Additionally, we obtained 200 manually designed rules from teachers that will be utilized in subsequent steps. As shown in Fig. 2(a), the elbow method analysis indicates the suitable number of clusters for this dataset is around 10. Based on the Xmeans algorithm, we set $K = 10$. Fig. 2(b) shows the t-SNE visualization of clustering results reduced to 2 dimensions.

Our dataset is sourced from Educoder,[2] a professional online education platform in collaboration with numerous universities and companies. The dataset includes multi-dimensional distributions of student information, grouping information, and group scores. It encompasses multiple instances of group discussions and assignments, with varying group sizes. Specifically, it comprises data for 3966 groups with a size of 2, 4167 groups with a size of 3, 1841 groups with a size of 4, 1759 groups with a size of 5, and 912 groups with a size of 6. We have processed the data by removing outliers and standardizing it, enabling meaningful comparisons across different instances of group assignments. In our experiments, our primary focus was on using groups with a size of 4. **Baseline:** We selected the random assignment strategy as the baseline. This random baseline is a simple and trivial approach that relies on arbitrary grouping decisions without utilizing any optimization algorithms or learning processes. Under this baseline, student groups are formed entirely randomly, without considering educational requirements, individual student differences, or other relevant factors.

DRL setting: We utilize reinforcement learning to select 20 rules out of the 200 for linear programming. A DQN with two hidden layers (each containing 128 units) was implemented in PyTorch for rule selection. The input state is a 200-dim binary vector representing currently selected rules. The output action is the rule ID to be selected. The reward consists of the linear programming runner result, repeating rule selection penalty and grouping failure penalty.

PSG evaluation method: We first initialized the algorithm framework for generating the initial grouping scheme in the experimental setup. Based on this initial policy the evolved policy, we evaluated its performance in the context of classroom teaching. During the teaching sessions, educators thoroughly reviewed and assessed the proposed grouping scheme based on criteria such as effectiveness, rationality, and alignment with practical needs. Subsequently, the educators provided corresponding score feedback. Scoring is based on the average of scores from multiple raters.

4.2. Comparative study

Comparison between different groups with different PSG category: To establish the foundation of the PSG method, we aimed to demonstrate the correlation between the composition of groups and their performance. Employing the PSG approach, we compared the score distributions of groups with different PSG types, as shown in Fig. 3(a). It is evident that groups with different PSG types exhibit distinct score distributions. Thus, it is feasible to enhance overall student performance by selecting appropriate group compositions. To validate the effectiveness of the scoring model, we utilized 80% of the data for training and the remaining 20% for testing. After 3000 epochs of training, the scoring model successfully predicted performance scores for different group types, as depicted in Fig. 3(b). This information can be utilized to guide the learning process of reinforcement learning models.

Comparison between two iterations: We trained for 200,000 steps on the unmodified 200 rules from teachers, and the reward trend is shown in Fig. 4(a). Subsequently, we manually evaluate the output and update rules using the evaluation. We continue training for 200,000 steps. The reward trend is shown in Fig. 4(b). It can be observed that after rules are modified, the reward drops in the short term but recovers after some training. Also, the second training converges faster than the first. Please notice that, due to the recent revision in scoring, we cannot directly compare the reward values. Consequently, our analysis will focus solely on observing the trend of the reward values.

Comparison between Random Grouping and PSG: By incorporating the reinforcement learning reward mechanism, PSG gradually learned and adjusted its strategies to achieve higher teacher ratings. Through multiple iterations of evaluation and adjustments, the reinforcement learning model continuously updated its policies based on educator feedback, resulting in improved grouping schemes. The comparison with the random baseline presented in Table 1 demonstrated the significant performance enhancement achieved by our reinforcement learning method, leading to superior grouping outcomes.

[2] https://www.educoder.net/.

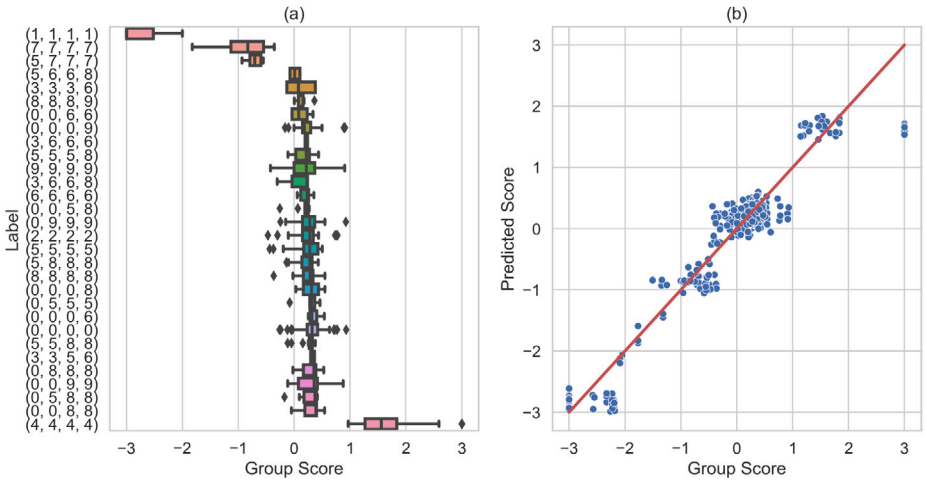

Fig. 3. PSG category comparison & score prediction model visualization.

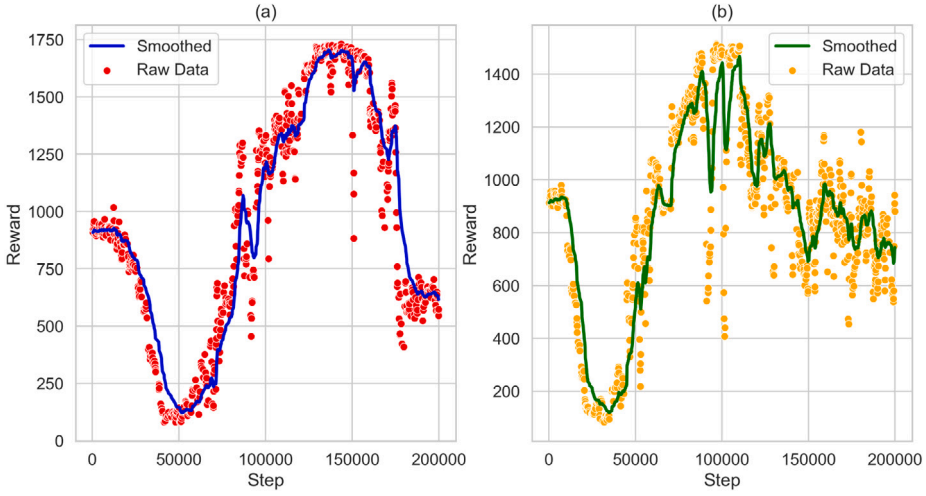

Fig. 4. Reward graph.

Table 1
Comparison between random grouping and PSG.

Iteration	Random grouping	PSG
Iteration 1	5	5
Iteration 2	5	5.5
Iteration 3	5	6
Iteration 4	5	7

4.3. Ablation study

Ablation Study of DRL: To demonstrate the necessity of the reinforcement learning part of the PSG algorithm, we conduct ablation experiments on MOOCs dataset. We measure the time it takes to generate a grouping proposal for a trained RL model. The baseline in the comparison is not using reinforcement learning but directly using linear programming to group according to rules. From the results in Table 2, it can be seen that the time cost grows slowly with dataset size increasing, demonstrating the computational efficiency of PSG. We conclude that efficiently acquiring a subset of rules via DRL for grouping improves efficiency and PSG can maintain good performance in large-scale student groupings.

Table 2
Ablation study of DRL.

Group size	Rules count	Without DRL	PSG
4	200	0.05 s	0.2 s
20	1000	1.93 s	0.5 s
100	10,000	27.76 s	1.2 s

5. Conclusion and future work

In this work, we propose PSG, which applies deep reinforcement learning and linear programming techniques for automated student group generation in collaborative learning environments. Students are first efficiently partitioned based on their multidimensional behavioral information using k-means clustering to generate personas-based clusters. A DRL agent then learns effective strategies by combining it with a linear programming solver to select personalized grouping rules. Experiments show that PSG achieves superior performance in student group formation compared to baselines in meeting pedagogical goals.

Future work might include trying different reinforcement learning algorithms, optimizing the reinforcement learning reward function, improving performance, and deploying it in real online classroom environments. We believe PSG can provide a new perspective on enhancing collaborative learning through intelligent student grouping.

CRediT authorship contribution statement

Shaojie Ma: Writing – original draft, Software, Methodology. **Yawei Luo:** Writing – review & editing, Supervision, Project administration. **Yi Yang:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] D.W. Johnson, R.T. Johnson, K.A. Smith, Cooperative Learning Returns To College What Evidence Is There That It Works? Change: Mag. Higher Learn. 30 (4) (1998) 26–35.

[2] C.J. Roseth, D.W. Johnson, R.T. Johnson, Promoting early adolescents' achievement and peer relationships: The effects of cooperative, competitive, and individualistic goal structures, Psychol. Bull. 134 (2) (2008) 223–246.

[3] D. Hayatpur, T. Helfenbaum, H. Xia, W. Stuerzlinger, P. Gries, Structuring collaboration in programming through personal-spaces, in: Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems, ACM, 2023, pp. 1–7.

[4] J. Hoffman, Flexible Grouping Strategies in the Multiage Classroom, Theor. Into Pract. 41 (1) (2002) 47–52.

[5] L. Silva, A.J. Mendes, A. Gomes, Computer-supported collaborative learning in programming education: A systematic literature review, in: 2020 IEEE Global Engineering Education Conference (EDUCON), 2020, pp. 1086–1095.

[6] R.S. Sutton, A.G. Barto, Reinforcement Learning, Second Edition: an Introduction, MIT Press, 2018.

[7] Y. Zheng, C. Li, S. Liu, W. Lu, An improved genetic approach for composing optimal collaborative learning groups, Knowl.-Based Syst. 139 (2018) 214–225.

[8] S. Xu, A.G. Zhang, S. Oney, How pairing by code similarity influences discussions in peer learning, in: Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, 2023, pp. 1–6.

[9] Y. Lou, P.C. Abrami, J.C. Spence, C. Poulsen, B. Chambers, S. d'Apollonia, Within-class grouping: A meta-analysis, Rev. Educ. Res. 66 (4) (1996) 423–458.

[10] M. Saleh, A.W. Lazonder, T.d. Jong, Structuring collaboration in mixed-ability groups to promote verbal interaction, learning, and motivation of average-ability students, Contemp. Educ. Psychol. 32 (3) (2007) 314–331.

[11] D. Chiu, Y. Beru, E. Watley, S. Wubu, E. Simson, R. Kessinger, A. Rivera, P. Schmidlein, A. Wigfield, Influences of math tracking on seventh-grade students' self-beliefs and social comparisons, J. Educ. Res. 102 (2) (2008) 125–136.

[12] P.K. Murphy, J.A. Greene, C.M. Firetto, M. Li, N.G. Lobczowski, R.F. Duke, L. Wei, R.M.V. Croninger, Exploring the influence of homogeneous versus heterogeneous grouping on students' text-based discussions and comprehension, Contemp. Educ. Psychol. 51 (2017) 336–355.

[13] N.M. Webb, G.P. Baxter, L. Thompson, Teachers' grouping practices in fifth-grade science classrooms, Elementary Sch. J. 98 (2) (1997) 91–113.

[14] S. Hooper, The effects of persistence and small group interaction during computer-based instruction, Comput. Hum. Behav. 19 (2) (2003) 211–220.

[15] H. Spoelstra, P. van Rosmalen, T. Houtmans, P. Sloep, Team formation instruments to enhance learner interactions in open learning environments, Comput. Hum. Behav. 45 (2015) 11–20.

[16] J. Moreno, D.A. Ovalle, R.M. Vicari, A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics, Comput. Educ. 58 (1) (2012) 560–569.

[17] L. Cen, D. Ruta, L. Powell, B. Hirsch, J. Ng, Quantitative approach to collaborative learning: performance prediction, individual assessment, and group composition, Int. J. Comput.-Support. Collab. Learn. 11 (2) (2016) 187–225.

[18] C.-T. Li, M.-K. Shan, Composing activity groups in social networks, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, 2012, pp. 2375–2378.

[19] I.U. Haq, A. Anwar, I.U. Rehman, W. Asif, D. Sobnath, H.H.R. Sherazi, M.M. Nasralla, Dynamic group formation with intelligent tutor collaborative learning: A novel approach for next generation collaboration, IEEE Access 9 (2021) 143406–143422.

[20] M. Chi, K. VanLehn, D. Litman, P. Jordan, Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies, User Model. User-Adapt. Interact. 21 (1) (2011) 137–180.

[21] A. Iglesias, P. Martínez, R. Aler, F. Fernández, Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems, Knowl.-Based Syst. 22 (4) (2009) 266–270.

[22] J.P. Rowe, J.C. Lester, Improving student problem solving in narrative-centered learning environments: a modular reinforcement learning framework, in: Artificial Intelligence in Education, Springer International Publishing, 2015, pp. 419–428.

[23] L. Panait, S. Luke, Cooperative multi-agent learning: The state of the art, Auton. Agents Multi-Agent Syst. 11 (3) (2005) 387–434.

[24] J. Bassen, B. Balaji, M. Schaarschmidt, C. Thille, J. Painter, D. Zimmaro, A. Games, E. Fast, J.C. Mitchell, Reinforcement learning for the adaptive scheduling of educational activities, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, 2020, pp. 1–12.

[25] S. Omidshafiei, D.-K. Kim, M. Liu, G. Tesauro, M. Riemer, C. Amato, M. Campbell, J.P. How, Learning to teach in cooperative multiagent reinforcement learning, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI Press, 2019, pp. 6128–6136.

[26] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678.

[27] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, Acm Comput. Surv. (CSUR) 31 (3) (1999) 264–323.

[28] P. Berkhin, A survey of clustering data mining techniques, in: Grouping Multidimensional Data: Recent Advances in Clustering, Springer, 2006, pp. 25–71.

[29] G. Gan, C. Ma, J. Wu, Data Clustering: Theory, Algorithms, and Applications, SIAM, 2020.

[30] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Oakland, CA, USA, 1967, pp. 281–297.

[31] M.E. Celebi, H.A. Kingravi, P.A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm, Expert Syst. Appl. 40 (1) (2013) 200–210.

[32] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii, Scalable k-means++, Proc. VLDB Endow. 5 (7) (2012) 622–633.

[33] D. Pelleg, A.W. Moore, X-means: Extending K-means with efficient estimation of the number of clusters, in: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., 2000, pp. 727–734.

[34] A.Y. Al-Omary, M.S. Jamil, A new approach of clustering based machine-learning algorithm, Knowl.-Based Syst. 19 (4) (2006) 248–258.

[35] H. Wang, Y. Yang, B. Liu, H. Fujita, A study of graph-based system for multi-view clustering, Knowl.-Based Syst. 163 (2019) 1009–1019.

[36] F. Cao, J. Liang, D. Li, L. Bai, C. Dang, A dissimilarity measure for the k-Modes clustering algorithm, Knowl.-Based Syst. 26 (2012) 120–127.

[37] T. Herawan, M.M. Deris, J.H. Abawajy, A rough set approach for selecting clustering attribute, Knowl.-Based Syst. 23 (3) (2010) 220–231.

[38] S. Zhu, D. Wang, T. Li, Data clustering with size constraints, Knowl.-Based Syst. 23 (8) (2010) 883–889.

[39] J. Ji, W. Pang, C. Zhou, X. Han, Z. Wang, A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data, Knowl.-Based Syst. 30 (2012) 129–135.

[40] R.S.J.d. Baker, K. Yacef, The State of Educational Data Mining in 2009: A Review and Future Visions, J. Educ. Data Min. 1 (1) (2009) 3–17.

[41] C. Romero, S. Ventura, P.G. Espejo, C. Hervás, Data mining algorithms to classify students, in: Educational Data Mining 2008, 2008.

[42] T. Barnes, The q-matrix method: Mining student response data for knowledge, in: American Association for Artificial Intelligence 2005 Educational Data Mining Workshop, AAAI Press, Pittsburgh, PA, USA, 2005, pp. 1–8.

[43] A.R. Anaya, J.G. Boticario, Application of machine learning techniques to analyse student interactions and improve the collaboration process, Expert Syst. Appl. 38 (2) (2011) 1171–1181.

[44] V. Klee, G.J. Minty, How good is the simplex algorithm, Inequalities 3 (3) (1972) 159–175.