Research Article

Improvement of AHMES Using AI Algorithms

Le Chen and JeongYoung Song

Department of Computer Science, Paichai University, West District, Daejeon 35345, Republic of Korea

Correspondence should be addressed to Le Chen; 1914717@pcu.ac.kr

Received 3 November 2021; Revised 10 December 2021; Accepted 21 December 2021; Published 10 January 2022

Academic Editor: Rahib Abiyev

Copyright © 2022 Le Chen and JeongYoung Song. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This research aims to improve the rationality and intelligence of AUTOMATICALLY HIGHER MATHEMATICALLY EXAM SYSTEM (AHMES) through some AI algorithms. AHMES is an intelligent and high-quality higher math examination solution for the Department of Computer Engineering at Pai Chai University. This research redesigned the difficulty system of AHMES and used some AI algorithms for initialization and continuous adjustment. This paper describes the multiple linear regression algorithm involved in this research and the AHMES learning (AL) algorithm improved by the Q-learning algorithm. The simulation test results of the upgraded AHMES show the effectiveness of these algorithms.

1. Introduction

When AHMES entered the end of the elaboration phase of RUP, we found that the existing difficulty management module and exam paper design module were flawed. In this improvement, we completely overturned the old design and redesigned it based on some AI algorithms.

We canceled the five difficulty levels of the mathematical model and quantified it with the help of machine learning algorithms. The AHMES_Learning reinforcement learning algorithm is designed to realize subsequent adjustment of the function of exam paper design and the difficulty of the mathematical model.

The goal of this improvement is to improve the rationality, accuracy, and intelligence of AHMES through AI algorithms to improve the performance of AHMES in long-term use. At the same time, this research also provides some ideas for using AI algorithms to improve other existing exam systems.

Section 2 introduces the difficulty management and the paper design of AHMES before this improvement. Section 3 introduces the tools and algorithms used in this improvement. Section 4 introduces the improved results and analysis. Sections 5 and 6 conclude and discuss this improvement.

2. Background

AHMES is a mathematical examination question generation system with a mathematical model as the core technology, and it is developed and maintained by the RUP development model [1]. Compared with the traditional examination system with a question bank as the core, AHMES generates mathematical questions from mathematical models. This makes AHMES have the characteristics of flexibility, scalability, and versatility. In the previous research, our team completed the main framework and functions of AHMES around the subject of linear algebra. This work proved the feasibility of mathematical models to generate mathematical problems. At the end of the elaboration phase, the AHMES project encountered some difficulties. These dilemmas led us to decide to redesign the exam paper design module and difficulty management module of AHMES. This section introduces the existing difficulty management and exam paper design schemes of AHMES and then discusses the difficulties encountered by AHMES.

2.1. Existing Difficulty Management Solution. In the current version of AHMES, the quantification of the difficulty of the mathematical model is calculated by

$$D = 1 - \frac{\bar{x}}{100}, \quad (1)$$

where $\bar{x}$ is the average score of all the students on this model.

According to the difficulty of the quantitative mathematical model, models are divided into five levels from easy to difficult. These five levels are described in Table 1.

Judging from the previous testing work, the five levels are too general. Although this method is more accurate, it seems that it is only suitable for online exams. In previous testing work, when users entered the exam results, they often only entered the average score and did not enter the average correct rate of each question. The reason is that, in offline exams, teachers manually or semimanually record the average correct rate of each question is almost impractical. We hope that, after this improvement, users only need to tell AHMES the average score of this exam after the exam is over and AHMES will do all other things.

2.2. Existing Exam Paper Design Solution. The exam paper design module designs exam paper structure and question type of appropriate difficulty according to the exam paper parameters' input by teacher.

The exam paper design module in the current version is a completely random selection of mathematical models under the specified difficulty. The difficulty of the questions in the exam papers generated by this method is fixed and that is not in line with the situation.

However, it is unreasonable that the difficulty of each question is the same on an exam paper. In the exam paper designed by hand, various difficult questions appear under the premise of estimating and controlling the average score of the paper. The existing solution does not have the ability.

2.3. Related Research. The traditional examination system quantifies the difficulty of questions based on the accuracy rate. When composing the test paper, select the corresponding question from the question bank according to the reference difficulty of the test paper. Qiong Chen's research has proposed an improved method for calculating the difficulty of examination questions [2]. In the previous version, AHMES referred to this method to quantify the difficulty of the mathematical model. Section 2.1 describes in the limitations of this method in AHMES.

We have tried to devise some methods to address these problems. Finally, we found that some ideas of Q_Learning are suitable for addressing the problems of AHMES. Prior to this, reinforcement learning has not previously been used in the examination system. Q_Learning is a value-based reinforcement learning algorithm [3]. Q_Learning selects the optimal strategy that can achieve its objectives through learning. However, Q-Learning cannot be used well in AHMES due to the flat characteristic of Q_Learning and the large number of mathematical models of AHMES.

3. Method

We decided to design an AI algorithm suitable for AHMES based on the idea of Q-learning. For this reason, the coefficient of difficulty needs to be redesigned and initialized. We reinitialized the difficulty coefficient of mathematical models using the multivariate linear regression method based on the existing historical data. And based on the Q_Learning algorithm, a reinforcement learning algorithm AHME-S_Learning, which is suitable for the AHMES is designed.

3.1. Tools. Python is still chosen as the development language to implement this improvement. NumPy, Pandas, StatsModels, and other modules in Python provide rich and convenient scientific computing methods for AI algorithms [4, 5]. In subsequent versions of AHMES, some algorithms may also be implemented with the help of TensorFlow [6] or PyTorch [7] to achieve better performance.

MongoDB joined AHMES for the first time in this research. MongoDB is a database system between relational and nonrelational. It provides a convenient way to store rich types of data. [8].

Github, as the version management tool, is still used in this research. We promise to open all codes of AHMES on Github in the future.

The hardware used for the calculation is Inter i7-9700K, 32 GB RAM, NVIDIA 2080ti.

3.2. Redesign of the Coefficient of Difficulty. A more detailed description of the degree of difficulty and coping with the improvement of this research are the main goals to be achieved by the new difficulty management solution. In addition, what we need to complete is the given initial difficulty coefficient of all mathematical models based on historical data.

3.2.1. Description of Difficulty Coefficient. To achieve the goal of this research, the difficulty of the mathematical model is no longer five levels but a quantifiable real number. We call this real number the difficulty coefficient of the mathematical model. Ideally, the value range of the difficulty coefficient is from 0 to 100. If the difficulty coefficient is 0, the correct rate of the questions generated by the mathematical model is 0, and the difficulty coefficient of 100 indicates that the correct rate of the questions generated by the mathematical model is 100.

However, some interesting and special data appeared in the follow-up testing work. Through analysis, we believe that the difficulty coefficient of the mathematical model not only reflects its correct rate but also some other influencing factors. Therefore, we believe that the difficulty coefficient may be negative or greater than 100. Section 4 introduces and analyzes these situations.

TABLE 1: Difficulty level of the models.

Value D	Level	Description
$D \leq 0.15$	1	Very easy
$0.15 \leq D < 0.4$	2	Easy
$0.4 \leq D < 0.65$	3	Common
$0.65 \leq D < 0.85$	4	Difficult
$D \geq 0.85$	5	Very difficult

We store the difficulty coefficients in MongoDB. Table 2 shows the logical structure of the table of difficulty coefficients' table.

3.2.2. Initialization of $D(m, d)$. To deal with the new solution, we need to requantify the difficulty of mathematical models with the existing historical data. Many historical records only have average scores without detailed correct rates for each question or mathematical model.

For the same students, the difficulty coefficient d_m of the mathematical model with serial number m is fixed. Formula (2) shows the relationship between the average score a and the difficulty coefficients d of mathematical models:

$$a_i = \gamma_i * \sum_{m=1}^t \frac{s_{im} * d_m}{\text{sum}_i}, \quad (2)$$

where i represents the exam with the serial number i and γ_i represents the score coefficient of the exam i . For example, $\gamma = F/100$ means an exam paper with the full score of F . t represents the total number of mathematical models in AHMES. s_{im} represents the number of test questions generated by the mathematical model m in the test i . sum_i represents the total number of questions in the test i .

Formula (3) shows the relationship between γ and the average score a of the corresponding exam and the correct rate r of all questions in the exam:

$$\gamma = \frac{a}{r * 100}. \quad (3)$$

From formula (2), it can be seen that a and d are linear relationships without constant term. The value of d can be obtained using the method of multivariate linear regression.

We separate usage of the mathematical model in each exam recorded in the history record into matrix S . The average score matrix A and the score coefficient matrix Y are calculated to obtain the correct rate matrix R . The relationship between matrix A , Y , and R refer to formula (3). Then, we establish matrix D to represent the difficulty coefficient of each mathematical model. Formula (4) shows matrix S , matrix D , and matrix R :

TABLE 2: Table of difficulty coefficients $D(m, d)$.

Mathematical model number	Difficulty coefficient
1	d_1
2	d_2
3	d_3
...	...
T	d_t

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1m} & \cdots & s_{1t} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{i1} & \cdots & s_{im} & \cdots & s_{it} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{k1} & \cdots & s_{km} & \cdots & s_{kt} \end{bmatrix},$$

$$D = \begin{bmatrix} d_1 \\ \vdots \\ d_m \\ \vdots \\ d_t \end{bmatrix}, \quad (4)$$

$$R = \begin{bmatrix} r_1 \\ \vdots \\ r_i \\ \vdots \\ r_k \end{bmatrix},$$

where i indicates the exam with number i , k is the total number of exam records, m represents the mathematical model with number m , and t represents the total number of all the mathematical models in AHMES.

Set D as the independent variable and R as the dependent variable. D is predicted by S and R . Equation (5) shows the relationship between S , D , and R :

$$R = S D. \quad (5)$$

In AHMES, the quantified difficulty of each mathematical model is an independent variable. That results in many independent variables in the linear model. The amount of calculation here is huge. Therefore, we choose to predict D through the OLS (ordinary least squares) regression algorithm in machine learning.

3.3. AHMES_Learning. AHMES_Learning (referred to as AL) is a reinforcement learning algorithm based on Q_Learning [3] and adapted to AHMES. AL provides AHMES with more

FIGURE 1: Flowchart of AL.

intelligent and reasonable exam paper design strategy and adjustment function through self-learning.

Figure 1 shows flowchart of AL.

Compared with the Q_Learning algorithm, the Q table in AL calculates the reward value based on $D(m, d)$ stored in MongoDB. Q table will be regenerated every time the test paper design function is executed. The reference difficulty of the current question is denoted by d' . s_r is the estimated average score entered by the user. e is a 0-1 random number. ε represents the strategy used to control randomness in decision-making. For example, when ε equals 0.9, it means that AL will choose the mathematical model according to the optimal value in 90% of cases and choose the random mathematical model 10% of cases.

Formula (6) is a strategy for selecting the mathematical model m . To reduce the occurrence of the same

mathematical model, we preset a repeat value. Whenever a mathematical model is selected, its value in the Q table will subtract the repeat value:

$$\begin{aligned}
 &\text{if } e < \varepsilon \text{ then :} \\
 &\quad \text{choose } m \text{ which } \max(-1 * |d' - q_m|) \\
 &\quad \text{else :} \\
 &\quad \text{randomly choose a model} \\
 &\quad q_m^- = \text{repeat.}
 \end{aligned} \tag{6}$$

The function of formula (7) is to recalculate the reference difficulty d' before each execution of the next round of model choosing strategy. sum is the number of questions entered by the user for this exam. c is the number of

FIGURE 2: Modified structure of AHMES.

questions which has been chosen a mathematical model. $d(m_k)$ is the difficulty coefficient of the k th selected mathematical model:

$$d' = \frac{1}{\text{sum} - c} * \left[s_r * \text{sum} - \sum_{k=1}^c d(m_k) \right]. \quad (7)$$

The function of formula (8) is to estimate the average score of the exam paper after all choosing work has been finished:

$$s_e = \gamma * \frac{1}{\text{sum}} * \sum_{k=1}^{\text{sum}} d(m_k). \quad (8)$$

After the exam is over, the user inputs the realistic average score s into the system. AHMES can perform the learning function to update $D(m, d)$. Formula (9) is the learning formula of AL algorithm. α is the preset learning rate to determine how much of the error is to be learned each time. The value range of α is 0 to 1:

$$\text{Repeat models: } d_m = d_m + \alpha \left[\left(\frac{s}{s_e} * d_m \right) - d_m \right]. \quad (9)$$

3.4. Modified Structure and Flow. AL changed the part of the core mechanism of AHMES. The structure and flow of AHMES needed to be changed accordingly. The details of the

design before the modification could be found in our previous paper “Development of AHMES (Automatic Higher Mathematics Examination System) Using Rational Unified Process.”

AI module and data module were utilized to replace exam paper design module and difficulty controller. AI module is an implementation of AHMES_Learning. It contains all the functions of the exam paper design module and the difficulty adjustment function in the difficulty controller. Data module includes the database and related functions of calling the database. In addition, the engine was redundant and finally abandoned because the selection of the model is no longer made by the engine in question-generating module but by AL. The optimization direction of the question generating module is flat. Figure 2 shows the structure of the improved AHMES.

We also adjusted the flow of AHMES because of the adjustment of the mechanism and the structure. The modification of the flow was focused around AI module and data module. In addition, it should be noted that the choosing model function of engine is implemented and replaced by AL. Figure 3 shows the improved AHMES process.

4. Results

4.1. Initial Difficulty Coefficient. AHMES owns 181 mathematical models. The OLS method was used to estimate the difficulty system of 181 mathematical models. Currently,

FIGURE 3: Modified flowchart of AHMES.

there are 740 exam records in the database. These exam records contain invalid records such as real student exam scores, initial technical test records, and test records of teachers. In the end, 614 exam records were screened.

The line graph Figure 4 shows the result of comparing the mathematical model difficulty coefficient calculated by the old method with the OLS estimation result.

The curves between the old coefficients of difficulty and new coefficients of difficulty are similar but not the same. This result shows that the estimation by OLS here seems credible.

4.2. Simulation Test. In this study, we set the learning rate α to 0.01.

We adopted the simulation tests because the tests of AL require a relatively large sample. We first assume a fixed test group (Agent). Agent meets the following conditions:

- (1) Agent has a stable correct rate when facing each mathematical model
- (2) Agent has a volatility of δ in the simulation test

The estimation set in the simulation tests is divided into 80 points. The value of δ is set to 0.12. The number of simulation tests is set to 200.

Figure 5 shows the result in the simulation tests.

The simulation results of the agent fluctuated greatly and irregularly before the 70 to 80 rounds of testing. However, the running result of the agent tends to be stable gradually in subsequent rounds. The result proves that AL is effective in the simulation test.

4.3. One Typical Experiment. This section shows one experiment. First, we entered the estimated average score of 80 points and the number of questions 10. Then, AHMES generated an examination paper. Table 3 lists the key data in the process of generating this examination paper.

AL triggered a random mechanism when generating the sixth question. A more complicated model appeared here, which led to a significant drop in the current estimated average score. To deal with this situation, AL chose the relatively simple models in the next two questions. There was not any random event again after generating the

FIGURE 4: Model difficulty curves.

FIGURE 5: Simulation test result.

TABLE 3: Data in operation.

Cycle number	Model	Difficulty coefficient	Current score
1	Determinant_value-29	80.1364	80.1364
2	Inverse_matrix_with_x2-4	77.3841	78.76025
3	Inverse_matrix-8	84.9474	80.82263333
4	Unknown_matrix_1-3	76.9874	79.863825
5	Algebraic_cofactor-17	85.7525	81.04156
6	unknown_matrix_2-10	36.1456	73.5589
7	operation_1-1	95.5981	76.70735714
8	operation_1-2	95.103	79.0068125
9	Linear_equations-2	88.0973	80.01686667
10	Inverse_matrix_with_x1-1	78.0947	79.82465

TABLE 4: Result of simulation and adjustment.

Model	Difficulty coefficient before adjustment	Difficulty coefficient after adjustment
Determinant_value-29	80.1364	80.1023
Inverse_matrix_with_x2-4	77.3841	77.3511
Inverse_matrix-8	84.9474	84.9112
Unknown_matrix_1-3	76.9874	76.9546
Algebraic_cofactor-17	85.7525	85.716
unknown_matrix_2-10	36.1456	36.1302
operation_1-1	95.5981	95.5574
operation_1-2	95.103	95.0625
Linear_equations-2	88.0973	88.0598
Inverse_matrix_with_x1-1	78.0947	78.0614

sixth question. The final estimated average score is 79.8. The agent conducted 40 simulations and got an average score of 76.4. AL adjusted the difficulty of these models based on this result. Table 4 lists the adjustment results.

5. Conclusion

This research proposes some AI methods suitable for AHMES. This paper designs a new difficulty system for AHMES and initializes it through OLS algorithm. This paper proposes the AL algorithm to achieve a more intelligent difficulty adjustment method. Simulation results show that these AI algorithms bring more rationality and intelligence to AHMES. The next step will be to study the improvement of the AL algorithm to reduce the amount of training required.

6. Discussion

The number of real samples is limited because AHMES has not yet been deployed in the real environment. We quoted simulation tests while samples of AI algorithms require a large size. Many tests of these AI algorithms were simulated because of the limitations of the test conditions. So, the results in the real environment need to be observed and followed up. These results will lead to appropriate adjustments to the parameters of these AI algorithms and even the possibility of redesign.

In addition, the current AI algorithms of AHMES are only suitable for a relatively fixed environment. For example, a university with a very high level of students and a university with a lower level of students need to setup two sets of AHMES. So, we are looking for solutions that can be applied to different levels at the same time.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

References

- [1] Le Chen, "Development of AHMES (automatical higher mathematics examination system) using rational unified process," *Mathematical Problems in Engineering*, vol. 2021, Article ID 7952816, 2021.
- [2] Q. Chen, "An application of online exam in discrete mathematics course," in *Proceedings of the ACM Turing Celebration Conference - China (TURC '18)*, pp. 91–95, Association for Computing Machinery, Hefei, China, May 2018.
- [3] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
- [4] T. E. Oliphant, "Python for scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 10–20, 2007.
- [5] W. Mckinney, "Python for data analysis: data wrangling with Pandas," *NumPy, and IPython*, 2017.
- [6] A. Géron, *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'reilly Media inc, Sebastopol, California, 2017.
- [7] N. Ketkar, "Introduction to PyTorch," *Deep Learning with Python*, Apress, Berkeley, California, CA, 2017.
- [8] K. Banker, *MongoDB in Action*, Manning Publications Co, 2011.