User-Adaptive Preparation of Mathematical Puzzles Using Item Response Theory and Deep Learning

Ryota Sekiya[1] Satoshi Oyama[12] Masahito Kurihara[1]

[1] Hokkaido University, Sapporo, Japan
[2]RIKEN AIP, Tokyo, Japan
r_sekiya@complex.ist.hokudai.ac.jp
{oyama, kurihara}@ist.hokudai.ac.jp

Abstract. The growing use of computer-like tablets and PCs in educational settings is enabling more students to study online courses featuring computer-aided tests. Preparing these tests imposes a large burden on teachers who have to prepare a large number of questions because they cannot reuse the same questions many times as students can easily memorize their solutions and share them with other students, which degrades test reliability. Another burden is appropriately setting the level of question difficulty to ensure test discriminability. Using magic square puzzles as examples of mathematical questions, we developed a method for automatically preparing puzzles with appropriate levels of difficulty. We used crowdsourcing to collect answers to sample questions to evaluate their difficulty. Item response theory was used to evaluate the difficulty of the questions from crowdworkers' answers. Deep learning was then used to build a model for predicting the difficulty of new questions.

Keywords: Computer-Aided Test, Item Response Theory, Crowdsourcing, Deep Learning, Magic Square.

1 Introduction

Recent advances in information technology have led to the introduction of computer-like tablets and PCs in educational settings, enabling the worldwide spread of educational systems that leverage the reach of the Internet and the power of computers.

BBC News reported that almost 70% of primary and secondary schools in the United Kingdom use tablet PCs and that 45% of the schools not currently using tablets have plans to introduce them in the future. Their introduction should be especially beneficial to students who have trouble studying using traditional methods [1]. Research by the Ministry of Education, Culture, Sports, Science and Technology in Japan revealed that there were about 1.5 million computers in elementary and junior high schools in Japan in 2018 and that whereas the number of students per computer was 8.4 in 2008, it was 6.4 in 2018 [2], so the introduction of computers into schools in Japan is steadily progressing.

In a computer-supported environment, students can study online courses featuring computer-aided tests (CATs) that are personalized to the student's skill level and learning style. Most CAT systems have an item pool for storing many questions from which questions suitable for each student are taken. Various patterns of questions with different levels of difficulty are required, especially in the mathematical area, because the skills of students tend to diverge, so the tests become meaningless if the students remember the questions. System creators must therefore prepare a large number of questions for each level of difficulty, and system administrators must frequently update the questions in the item pool so that students cannot answer the questions without thinking. Both tasks are burdensome.

Another problem with CATs is setting the level of question difficulty. The difficulty of a question is normally determined by considering the accuracy rate for the test, the number of hints or types that will be given, and experiences on this area. But these measures depend on various factors such as the skill and experience of the test creator and the overall skill level of the target group of students. Therefore, if the test creator or student group change, the scores on different tests cannot be directly compared. In such cases, a large number of students is required for each test to avoid degrading test reliability.

Using magic square puzzles as examples of mathematical questions, we developed a user-adaptive method for automatically selecting puzzles with appropriate levels of difficulty. We used crowdsourcing to collect answers for sample questions and used the answers to estimate the difficulty of the questions on the basis of item response theory. We then used machine learning to build a model for predicting the difficulty of new questions. Finally, by using these difficulties and students' skills as measured using a computer adaptive test, we developed a system for recommending questions that can improve the skill of students.

2 Related Work

Many researchers have considered automatic generation of educational questions in various subject areas. Hoshino and Nakagawa [3] focused on the English 4 choice question and used machine learning to identify places in sentences that could be blanked in order to make fill-in-the-blank questions. Hill and Simba [4] generated blank locations and distractors in multiple-choice fill-in-blank questions by calculating word co-occurrence likelihood using n-grams. Sakaguchi et al. [5] used a large number of sets of English sentences given by English learners and journals and found the error correction pairs and the misuse probabilities of words. They then used them and a support vector machine method to generate challenging distractors to confuse the student when choosing an answer. Liu et al. [6] generated questions in Chinese about given Chinese sentences by extracting and ranking five elements from the sentences: when, where, what, which, and who. Rocha and Faron-Zucker [7] and Papasalouros et al. [8] selected questions automatically from a database or a web by using strategy estimation based on domain ontology. Takano and Hashimoto [9] and Furudate et al. [10] created questions automatically by using a knowledge base. Takano and Hashimoto used a knowledge

base built in advance to store specific questions while Furudate et al. built the knowledge base itself by using morphological analysis and knowledge patterns to extract previous questions.

Most of these studies used databases, knowledge bases, and ontologies created by relevant experts, so burdens are placed on the experts to build the system and on administrators to update the system. While many related studies used natural language and knowledge obtained from the Web or journals, few studies have used automatic generation of mathematical questions with a large number of various patterns.

3 Item Response Theory (IRT)

Conventional test methods measure the skill of students on the basis of the total points or standard deviation values for a test. The results of such measures are often affected by such factors as uneven test difficulties among test creators or uneven skill levels among different groups of students. This makes is hard to compare the skills of students who take different tests and to compare the difficulties of questions answered by different groups of students.

Item response theory (IRT) is a method for probabilistically estimating latent parameters like the skill levels of students and the difficulties of questions simultaneously from the discrete responses of students to questions. Using IRT makes it possible to estimate parameters universally without the effects of the above-mentioned dependences. This enables comparison of parameter values estimated from different tests and groups of students on the same scale. IRT was derived from Lord's Theory of Test Scores [11].

IRT considers the probability p that student i with skill level θ can answer question j with difficulty b. This process is assumed to follow a logistic regression model with item discrimination a and approximation constant D:

$$p_j(\theta_i) = \frac{1}{1+\exp(-Da(\theta_i - b_j))}.$$

The larger the θ, the higher the skill level of the student; and the larger the b, the more difficult the question. The likelihood of a student correctly answering a question was calculated using function L,

$$L(u_i|\theta_i) = \prod_{j=1}^{n} p_j(\theta_i)^{u_{ij}}(1 - p_j(\theta_i))^{1-u_{ij}}.$$

where u_{ij} represents the response of student i to question j: $u_{ij} = 0$ means that the response was incorrect, and $u_{ij} = 1$ means that it was correct. These are the only observable responses in the model; the other parameters must be estimated. There are similar IRT models, but these models contain too many parameters and are too complex for our purpose. Therefore, we decided to use the above simple model.

We cannot calculate likelihood function L because we cannot observe θ as in a normal logistic regression. To estimate the parameters, we use a two-step algorithm proposed by Mislevy [12] that marginalizes θ and uses Bayesian inference to maximize

the likelihood function, which is calculated in a manner similar to that for an expectation-maximization (EM) algorithm.

4 Estimating the Difficulties of Magic Square Puzzles

We used supervised machine learning to prepare many magic square puzzles. Magic square puzzles are a kind of the mathematical puzzles that ask users to guess appropriate integers for some blank cells. In this section, we explain how we calculated the difficulties of these puzzles by using IRT and EM-type IRT models [13] to handle incomplete answer matrices, where students may not have answered all questions and questions may not have been answered by all students. To gather answer data, we used the Lancers crowdsourcing service [14], a service for obtaining input from many people through the Internet. Along with gathering the answer data, we gathered the time it took for the crowdsourcing worker to input an answer, his or her confidence that the answer was correct, and his or her subjective evaluation of the question's difficulty.

We created 330 magic square puzzles and obtained 9059 answers from 448 crowdsourcing workers. We estimated the difficulty of each puzzle by using the IRT model and EM-type IRT method. In addition to solving the puzzles, we also asked the workers to report other factors, as mentioned above: time needed to solve each puzzle, confidence of solution correctness, and subjective difficulty of each puzzle. In addition to the IRT-based difficulty, we also considered the time needed to answer a question by using a computer algorithm and the simple correct rate (rate of correct answers by workers) as measures of puzzle difficulty.

Table 1. Correlations between difficulty measures and other factors

	Time to complete cells	Confidence	Difficulty(subjective)	No. of blank cells
Time with computer algorithm	0.25	-0.24	0.20	0.22
Simple corrective rate	0.87	-0.98	0.95	0.90
Difficulty (IRT)	0.88	-0.97	0.96	0.90

Table 1 shows the correlations between the difficulty measures and the other factors. The top row shows the correlation for the time needed to solve a puzzle by computer using a backtracking algorithm. The middle row shows the correlation for the rate of correct answers by the workers. The bottom row shows the correlation for the difficulty estimated using the IRT. The correlations between the difficulty estimated using the IRT and the other factors were higher than those between the time needed by computer and those factors. This indicates that puzzle difficulty estimated by computer differs greatly from that estimated by people. The finding that the difficulty estimated using the IRT and the simple correct rate show similar correlations with other factors indicates that the differences in skill levels among the workers were relatively small.

From these results, we concluded that the difficulties of the puzzles estimated using the IRT were reliable and thus used them as the true difficulties in the supervised machine learning described in the next section.

5 Difficulty estimation using machine learning

Although the cost of crowdsourcing is relatively low, asking workers to answer questions to be used in a test is not realistic. Therefore, we used puzzles with various IRT-estimated levels of difficulty as training data and used supervised machine learning to build a model for predicting the difficulty of new puzzles. To evaluate the effectiveness of our approach, we compared the difficulties of puzzles as estimated by machine learning with those estimated by the IRT.

5.1 Model overview

In a preliminary study, we were unable to accurately estimate the difficulties of puzzles by using a simple neural network model, so we first trained three base models and then trained a combined model using the outputs of the three models. Fig. 1 shows the architecture of our combined model.

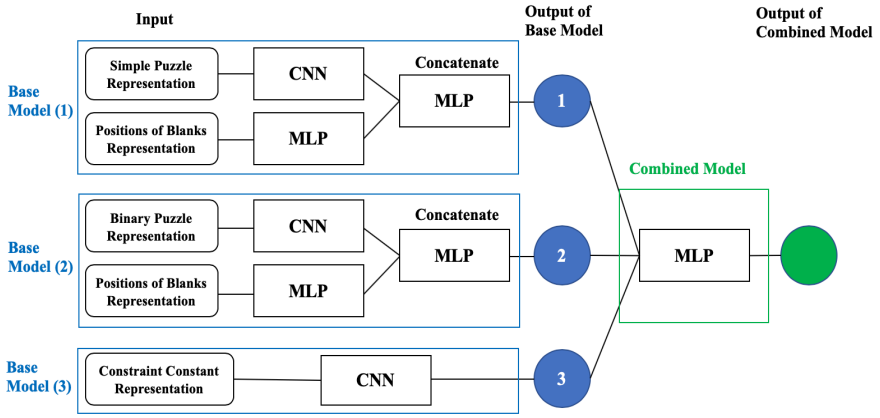

Fig. 1. Architecture of combined model

As inputs to the base models, we used different feature representations of a puzzle: simple puzzle, binary puzzle, positions of blank cells, and constraint condition. We represent the cell in $N \times N$ puzzle X at row i and column j as x_{ij} and the cell in feature matrix Y at row u and column v as y_{uv}. The simple puzzle representation represents a puzzle as an $N \times N$ two-dimensional array, with the puzzle being represented as it is except that when x_{ij} is blank, y_{uv} is filled with a zero. In binary puzzle representation, Y is an $N \times N \times N^2$ three-dimensional array. If the $x_{ij} = a$, a^{th} element of the array of

the third dimension is set to one, all the other elements of that array are set to zero. If x_{ij} is blank, all the elements in the corresponding array are set to zero. For example, if $x_{ij} = 11$, the array for the element is [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0]. The representation of positions of blank cells are represented using a one-dimensional array with N elements. When x_{ij} is a number, the corresponding element in the array is set to zero, and when x_{ij} is blank, the element is set to one. The constraint constant representation uses a $(2N + 2) \times N$ two-dimensional array in which each element represents a row, column, diagonal constraint in the original puzzle.

We trained three base models: (1) a combination of a convolutional neural network (CNN) for which the input is the simple puzzle representation and a multi-layer perceptron (MLP) for which the input is the representation of positions of the blank cells, (2) a combination of a CNN for which the input is the binary puzzle representation and a MLP for which the input is the representation of positions of the blank cells, and (3) a single MLP for which the input is the constraint constant representation.

5.2 Model evaluation

Fig. 2 shows the relationship between the ML-based difficulty and the IRT-based difficulty. Table 2 shows correlations between the two difficulties and other factors.

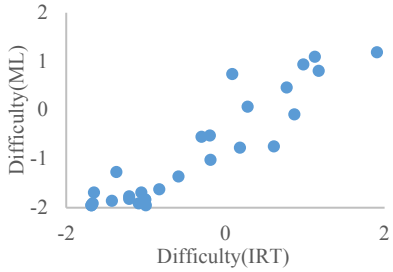

Fig. 2. ML-based difficulty against IRT-based difficulty

Table 2. Correlations between estimated difficulty and other factors

	Time for complete cells	Confidence	Difficulty(subjective)	No. of blank cells
IRT-based difficulty	0.90	-0.96	0.95	0.92
ML-based difficulty	0.88	-0.92	0.92	0.91

In Fig. 2, we can see the strong positive correlation (0.93) between the ML-based difficulty and IRT-based difficulty. In Table 2, we can see that the ML-based difficulty has as strong correlation with other factors as the IRT-based difficulty has. This result shows the learned model can accurately predict the IRT-based difficulty.

Table 3 shows the metrics for the combined model and the three base models. Max and Min are the maximum and minimum values between the true and estimated values; Max represents the positive difference, and Min represents the negative difference.

RMSE is the root mean square error of each model; the smaller the value the better. RCC is Spearman's rank correlation coefficient, which is used to evaluate the similarity of the ranking orders of two values. The larger its value, the more accurate the prediction of which of two puzzles X and X' is more difficult. RCC (blank cells) is the average of RCCs for each number of blank cells. The larger this value, the more accurate the model can distinguish the difficulty of puzzles with the same number of blank cells.

Table 3. Metrics for each model.

	Combined model	Base model (1)	Base model (2)	Base model (3)
Max	0.65	1.14	1.35	1.14
Min	-1.35	-1.62	-1.32	-1.79
RMSE	0.55	0.65	0.55	0.67
RCC	0.89	0.79	0.84	0.84
RCC (blank cells)	0.30	0.50	0.00	0.25

Base model (1) achieved a large RCC (blank cells), but its RCC was small and its RMSE was large. This model can distinguish the questions with the same number of blank cells, but overall accuracy is not good. Base model (2) achieved a good RMSE, but its RCC (blank cells) was small and the difference between Max and Min was very large. This model cannot distinguish the difference in difficulty among questions with the same number of blank cells, and sometime outputs very large outliers despite the overall high accuracy. Compared with the base models, the combined model inherits some strong points of the base models and can accurately rank puzzles based on their difficulties.

6 Conclusion and Future Work

We have developed a method for automatically selecting puzzles with appropriate levels of difficulty as a basis for the automatic preparation of mathematical questions, which will reduce the burden on question creators and system administrators. First, we calculated the numeric difficulties of mathematical questions by using item response theory and EM-type IRT, and then compared the values to those of other measures by using crowdsourced data. Next, we used machine learning to create a combined neural network model for estimating the difficulty of new questions. Testing showed that our combined model outperformed the three base models.

Future work includes developing a recommendation system that provides suitable questions to students on the basis of their skill levels as estimated using the IRT. It also includes building a machine learning model for predicting other values, like the time needed for a specific student to answer a specific question. Although we focused on only magic square puzzles, this method can be applied to other mathematical questions. We will thus work on expanding it for such applications and explore methods that support user-adaptive recommendation of questions.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Numbers JP15H02782 and JP18H03337, and by the Telecommunications Advancement Foundation.

References

1. Coughlan, S.: "Tablet computers in '70% of schools'", BBC NEWS, 3 December 2014, https://www.bbc.com/news/education-30216408.
2. Ministry of Education, Culture, Sports, Science and Technology: "The research for situation of educational informatization in school in 2017", https://www.e-stat.go.jp/stat-search/files?page=1&cycle_facet=cycle.
3. Hoshino, A., Nakagawa, H.: "A real-time multiple-choice question generation for language testing – a preliminary study", Proceedings of the 2nd Workshop on Building Educational Applications, Using NLP, pp. 17-20 (2005).
4. Hill, J., Simba, R.: "Automatic Generation of Context-Based Fill-in-the-Blank Exercises Using Co-occurrence Likelihoods and Google n-grams", Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 23-30 (2016).
5. Sakaguchi, K., Arase, Y., Komachi, M.: "Discriminative Approach to Fill-in-the-Blank Quiz Generation for Language Learners", Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 238-242 (2013).
6. Liu, M., Rus, V., Liu, L.: "Automatic Chinese Factual Question Generation", IEEE Transactions On Learning Technologies, Vol. 10, Issue 2, pp. 194-204 (2017).
7. Oscar Rodriguez Rocha, Catherine Earon Zucker: "Automatic Generation of Educational Quizzes from Domain Ontologies", EDULEARN 2017-9th International Conference on Education and New Learning Technologies, pp. 4024-4030 (2017).
8. Papasalouros, A., Kanaris, K., Kotis, K.: "Automatic Generation of Multiple Choice Questions From Domain Ontologies", Proceeding of the 8th International Conference on Web Intelligence, Mining and Semantics, Article No. 32 (2018).
9. Takano, A., Hashimoto, J.: "Drill Exercise generation based on the knowledge base", The Special Interest Group Technical Reports of Information Processing Society of Japan, NL-160, pp. 23-28 (2003).
10. Hurudate, M., Takagi, M., Takagi, T.: "A Proposal and Evaluation on a Method of Automatic Construction of Knowledge Base for Automatic Generation of Exam Questions", The Special Interest Group Technical Reports of Information Processing Society of Japan, Vol. 128, No. 14 (2015).
11. Lord, F.M.: "A theory of test scores (psychometric monograph)", Psychometric Society (1952)
12. Mislevy, R.J.: "Bayes model estimation in item response theory", Psychometrika, Vol. 51, Issue 2, pp. 177-195.
13. Sakumura, T., Tokunaga, M., Hirose, H.: "Making up the Complete Matrix from the Incomplete Matrix Using the EM-type IRT and Its Application", The Information Processing Society of Japan Journal Transactions on Mathematical Modeling and its Applications, Vol. 7, No. 2, pp. 17-26 (2014).
14. Lancers Homepage, https://www.lancers.jp.