Determination of Window size of Hidden Markov Item Response Theory

1831133

Boungnong Nilandone

Graduate School of Informatics and Engineering

The University of Electro-Communications

Table of Contents

Table of Figures

1. Introduction

In recent years, learning assistance has been gaining more attention in the education field. Because over-instruction or under-instruction can lead to ineffective knowledge development, determining the amount of support that a learner needs has been a major challenge for educators. Vygotsky (1962) introduced the Zone of Proximal Development (ZPD) for problem solving, where a learner cannot solve difficult tasks alone but can do so with an expert's help, thereby promoting learner development [1][2]. Using the ZPD concept, Wood et al. (1976), Collins (1989), and Bruner (1996) have shown that when learners face higher-level tasks, the teachers should provide moderate support depending on the learner's ability through the process of "scaffolding" [3][4][5]. Scaffolding is a process where the learners obtain support to solve tasks that are beyond their capability when solving by themselves. In the scaffolding process, the learner's knowledge will be measured, and after the teacher's assistance, the performance of the learners will be estimated. In other words, to effectively assist the learner, their knowledge and their performance must be accurately estimated.

To estimate the learner's knowledge, Ueno and Matsuo [6] and Ueno and Miyazawa [7][8] proposed the use of Item Response Theory (IRT). IRT is one of the test theories that can be used to estimate the learner's ability based on past learning data and can also be used to predict the response of the learner by calculating the probability of getting a correct answer based on the learner's estimated ability [9]. However, IRT assumes that each task is dependent on a static learner's ability, meaning that the learner's ability does not change during the learning process, which might lead to inaccurate prediction of the learner's response.

To handle the change in the learner's ability during the learning process, Tsutsumi et al. (2019) [10][11] proposed the Hidden Markov Item Response Theory (HMIRT) model, which treats the learner's ability as a time-series. HMIRT assumes that at some point during the learning process, the learner will gradually forget about past tasks. HMIRT uses the Sliding Window method to model the learner forgetting about the earlier tasks. HMIRT also assumes that the learner’s ability to perform each task does not change before the point at which the learner forgets, meaning that these tasks will be dependent on one value for the learner’s ability, the same as in the traditional IRT. After a learner increases his/her ability due to the learning effect, the learner's ability will be updated and used in the next task. To handle this process, HMIRT introduces two new parameters: the window size parameter is a fixed number used to control how many of the previous tasks affect the estimation of the learner’s current ability, and the variance parameter is a fixed number used to control the magnitude of change in the learner’s ability at each time point. This model fixes the problem of static ability in the traditional IRT model, leading to more accurate estimation of learner’s ability and therefore performance.

It has been shown that HMIRT estimates the learner’s ability better than the traditional IRT [10][11]. However, HMIRT’s constant window size might not guarantee an accurate estimation of learner’s ability. Another limitation of HMIRT is the fixed variance parameter. Setting a fixed variance parameter limits the change in the learner’s ability at each time state. If the variance parameter is small, the learner’s ability will not change much. If the variance parameter is large, the learner’s ability will change too much. Because the content of each task varies, the degree of understanding gained by completing each task must also be different. Therefore, accurate prediction cannot be guaranteed when using a fixed variance parameter. To solve these problems, we propose the Auto-

Fluctuation Window Size of Hidden Markov Item Response Theory Model. In this model, the window size and variance parameters are time series rather than fixed values so that the parameters can change at each time point. With this proposed model, we expected a more flexible and more accurate estimation of learner's ability.

2. Item Response Theory

To effectively support the learner's development, learner performance prediction is needed. To predict a learner's performance, Item Response Theory (IRT)[9][12] has been used. IRT is one of the test theories based on mathematical models and has been used widely in computer testing. It has the following advantages:

1. It is possible to assess ability while minimizing the effect of the heterogeneous or aberrant items, which has a low estimation accuracy.
2. The learners' responses to different items can be assessed on the same scale.
3. Missing data can be readily estimated.

In the IRT model, one of the most used models is the two-parameter logistic model (2PL). In the dichotomous response, x_{ji} denotes the response of the learner $j(1,..,n)$ to i-th item as:

$$x_{ji} = \begin{cases} 1\text{: correct response for } i\text{-th item} \\ 0\text{: incorrect response for } i\text{-th item} \end{cases}$$

With the learner's ability variable θ_j, 2PL can be expressed by:

$$P\left(x_{ji} = 1 | \theta_j, a_i, b_i\right) = \frac{1}{1 + exp\left\{-1.7a_i\left(\theta_j - b_i\right)\right\}} \qquad (1)$$

where the item parameter a_i and b_i is called the discrimination parameter and difficulty parameter, respectively, θ_j is the latent ability variable of learner j. The item parameter a_i, b_i was estimated in advance from the training data.

In this model, because all of the items depend on one prior distribution of ability variable, the estimation of the ability variable is less affected by the prior distribution but is easily affected by the learning process. Therefore, the over-training occurs and the ability variable might be overly estimated or underestimated.

In order to avoid the over-training, Tsutsumi et al. [10][11] proposed the Hidden Markov model, which changes the learner's ability to time-series where the current ability variable depends on the value of previous ability variable. With this model, the accuracy of the learner's ability estimation has been improved.

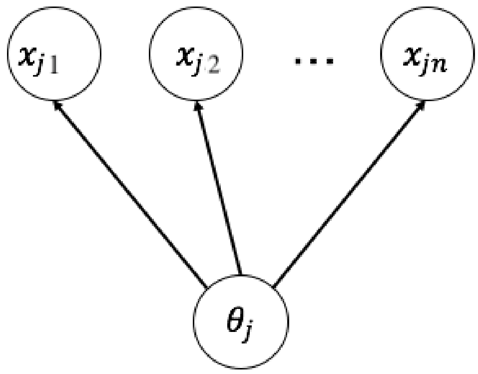

Fig 1 Traditional Item Response Theory model

3. Hidden Markov Item Response Theory

The Hidden Markov Item Response Theory (HMIRT) model is an extension of the IRT model that replaces the fixed value for the learner's ability θ_j with the time-series θ_{jt}, where the change in ability at time t depends on the value of the ability variable θ_{jt-1} at time $t-1$ according to a Hidden Markov process. Here, the number of task items used in the ability estimation at time t has been set, denoted by L. HMIRT assumes that the value of the ability variable does not change for items $i = 1, \dots,$ L, which means that these initial items will depend on the same ability value (as in the IRT model). When the item i > L, the ability variable θ_{jt} will change based on θ_{jt-1}. The variance parameter δ must be estimated to control the transition (amount of variation) of the ability variable θ_{jt} between each time state.

The transition model for the ability variable $\theta_{jt}(t = 1, \dots, I - L)$ uses the sliding window method [13][14]. The sliding window is a method of determining the number of hidden variables that will affect the ability estimation when shifting by the set window size. When the current item i > L, the ability estimation is conducted by shifting the window along the items one at a time (Figure 2).

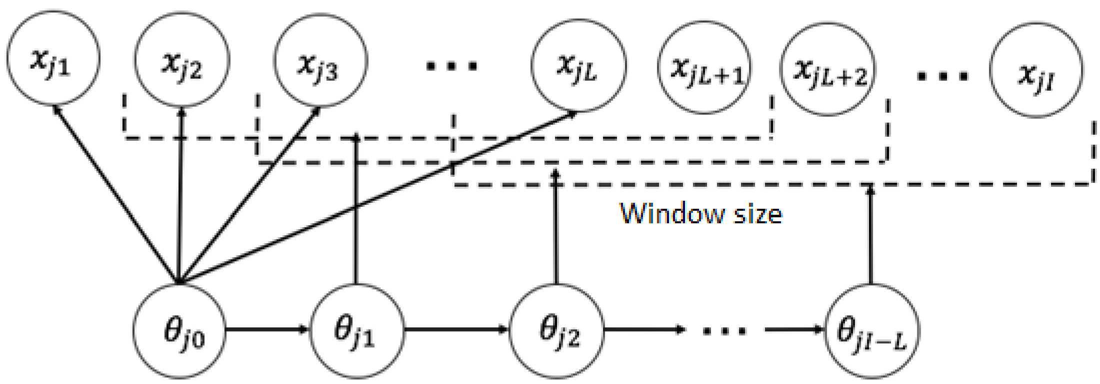

Fig 2 Representation of Hidden Markov Item Response Theory model.

In this model, the number of items that depends on one ability variable in each learning process is defined by the window size parameter L. The learning process at time t is as follows:

$$\begin{cases} t = 0: & i = 1, \dots, L \\ t = 1: & i = 2, \dots, L + 2 \\ \vdots & \vdots \\ t = I - L: & i = I - L, \dots, I \end{cases} \tag{2}$$

When L is small, only the learner's most recent history will influence their estimated ability θ_{jt}. If L is larger, additional task items will factor into the ability estimation.

This model was originally developed for the dynamic assessment system, which gives hints to the learners when they cannot solve the task. In this research, we generalize the model so that it can work without the hint. The probability P_{ijt} of a correct answer for task item i being provided by learner j based on their ability θ_{jt} at time t is as follows:

$$P_{ijt} = \frac{1}{1 + exp\left(-a_i(\theta_{jt} - b_i)\right)} \tag{3}$$

where

$$\theta_{jt} \sim N(\theta_{jt-1}, \delta) \tag{4}$$

$$\theta_{j0} \sim N(0,1) \tag{5}$$

δ is the variance parameter, which controls how much the estimated ability can change during each learning session. In this model, the window size parameter L and the variance parameter δ perform important roles in the prediction of the learner’s performance.

However, HMIRT’s assumption that the window size is fixed may not guarantee an accurate estimation of learner’s ability, since the previous tasks that contribute to the ability estimation at each time state can vary for the current task. Moreover, setting a fixed variance parameter limits the range of transition of the learner’s ability at each time state. To solve this problem, the Auto-Fluctuation Window Size of Hidden Markov Item Response Theory model has been proposed.

4. Auto-Fluctuation Window Size HMIRT

In previous research [10][11], it has been shown that the response prediction of HMIRT is more accurate than that of traditional IRT. However, by observing the prediction accuracy rate for each item, we found that some of the predictions of traditional IRT, specifically those where the window size parameter L is equal to the item number, are more accurate than those in HMIRT. With this fact, we can assume that in some cases, changing the window size can lead to a more accurate estimation of learner's ability. Moreover, the fixed variance parameter in HMIRT limits the range of transition of the learner's ability at each time state. Because the content of each task varies, the degree of understanding gained during that task must also be different. Therefore, making the variance parameter changeable at each time point can lead to a more accurate learner's ability estimation. To handle the changes in the window size and variance parameters at each time state, we propose the Auto-Fluctuation Window Size HMIRT (AFHMIRT) model.

The AFHMIRT model replaces the fixed values for the window size parameter L and variance parameter δ with the time-series window size L_t and variance parameter δ_t where t is the time state of the learning process. The model then estimates the window size L_t and the variance δ_t that maximize the response prediction accuracy for each item. In the response prediction process of the HMIRT model, the system first estimates the item parameters, then estimates the learner's ability for all of the time states $\boldsymbol{\theta}_j$, then finally calculates the prediction accuracy. Because we want to find the optimal window size and variance for each item, we need to calculate the prediction accuracy of each item. To be more precise, the proposed model will estimate the item parameters and the learner's ability for only the current time state, then calculate the response prediction accuracy for one item at a time while adjusting the window size and the variance to find the optimal window size for that item. When adjusting the window size and the variance, the model needs to re-estimate the item parameters and the learner's ability because these changes affect the calculation of the likelihood that will be used in parameter estimation. After re-estimating the parameters, the response prediction accuracy is re-calculated. The learning process at each time state can be written as follows:

$$\begin{cases} t = 0: & i = 1, \dots, L_0 \\ t = 1: & i = 2, \dots, L_1 + 2 \\ \vdots & \vdots \\ t = I - L_t: & i = I - L_t, \dots, I \end{cases} \tag{6}$$

Figure 3 is an example of how the model will look when obtaining the optimal window size parameter for each item. For a 7-item model, L_t={2,2,2,3,3,2}.

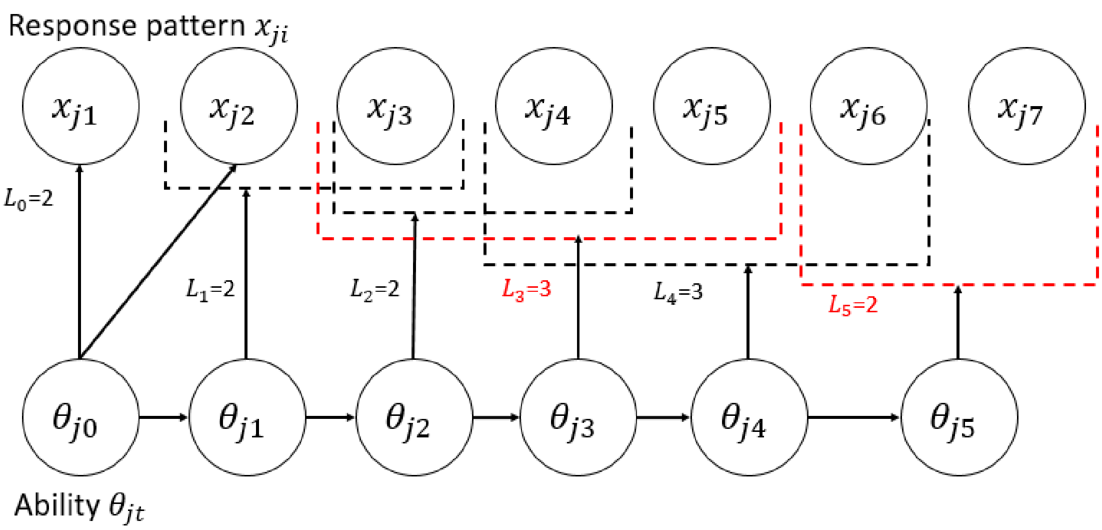

Fig. 3 Example of an Auto-Fluctuation Window Size HMIRT model

5. Parameter Estimation

One of the popular methods for estimating item parameters for the IRT model is to use the expectation-maximization (EM) and Newton-Raphson algorithms to estimate the marginal maximum likelihood (MML). The other method is maximum a posteriori (MAP) estimation. For both MML and MAP estimation, when the method is applied in a simple model such as a two-parameter logistic model or a grade response model, or when the dataset is large, the parameter estimation will be stable and accurate. On the other hand, when dealing with a complex model or when the dataset is small, the accuracy of the parameter estimation will be decreased. In recent years, the use of the Markov Chain Monte Carlo (MCMC) method to estimate the expected a posteriori (EAP) for parameter estimation has become more common. The MCMC method generates a random sample from the parameter's posterior distribution and uses the generated sample to estimate the parameter's expected value. In this research, we decided to use the MCMC method for parameter estimation because this method is better suited to the limited dataset and more complex model. In MCMC, there are many methods of generating a random sample; in this research, we use Metropolis-Hastings within a Gibbs algorithm. With the parameter $\theta = \{\theta_{10}, \dots, \theta_{jI-L}\}, a = \{a_1, \dots, a_I\}, b = \{b_1, \dots, b_I\}$ and the prior distribution $g(\theta_{jt}|\delta_t)$, $g(a_i)$, $g(b_i)$, given the response pattern X, the posterior distribution of the parameters can be expressed as follows:

$$p(\theta, a, b|X) \propto$$

$$L(X|\theta, a, b)g(a)g(b)g(\theta)$$

$$= \left[\prod_{t=0}^{I-L} \prod_{i=t+1}^{L+t+1} (P_{ijt})^{x_{ij}} (1 - P_{ijt})^{1-x_{ij}}\right] \left[\prod_{i=1}^{I} g(a_i)g(b_i)\right] \left[\prod_{t=0}^{I-L} \prod_{j=1}^{J} g(\theta_{jt})\right] \tag{7}$$

where

$\operatorname{Log} a_i \sim N(0.0, 0.2)$

$b_i \sim N(0.0, 1.0)$

$\theta_{j0} \sim N(0.0, 1.0)$

$$\theta_{jt} \sim N(\theta_{jt-1}, \delta_t) \tag{8}$$

Let θ'_j be the current parameter value for $\theta_{jt} = (\theta_{j0}, \ldots, \theta_{jI-L})$ and θ_j be a new proposal for the parameter obtained by the following:

$$\theta_j \sim N(\theta'_{jt-1}, 0.01) \tag{9}$$

The acceptance rate for the parameter sampling is then as shown below:

$$\alpha(\theta_j|\theta'_j) = \min\left(\frac{L(X_j|\theta_j, a', b') \prod_{t=0}^{I-L} g(\theta_{jt})}{L(X_j|\theta'_j, a', b') \prod_{t=0}^{I-L} g(\theta'_{jt})}, 1\right) \tag{10}$$

The same formula is applied for parameter sampling of a_i and b_i. .

In this research, we set the MCMC maximum chain length to 40,000 iterations. To eliminate the effect of the initial value, we set a burn-in period of 20,000 iterations. After the burn-in period, a sample is collected for an interval of 1000 iterations, and the average is taken to be the EAP estimation value. Pseudo-code for the parameter estimation is shown in Algorithm 1.

Algorithm 1 Parameter Estimation with MCMC

Given maximum chain length S,burn-in B,interval E
Initialize MCMC sample $A \leftarrow \emptyset$
Initialize θ^0, a^0, b^0
for $s = 1$ to S **do**
    for $j \in \{1, \ldots, J\}$ **do**
        Sample $\theta_j^s \sim N(\theta_j^{s-1}, 0.01)$
        Accept θ_j^s with the probability $\alpha(\theta_j|\theta'_j)$
    end for
    for $i \in \{t+1, \ldots, t+1+L_t\}$ **do**
        Sample $a_i^s \sim N(a_i^{s-1}, 0.01)$
        Accept a_i^s with the probability $\alpha(a_i|a'_i)$

```
        Sample $b_i^s \sim N(b_i^{s-1}, 0.01)$
        Accept $b_i^s$ with the probability $\alpha(b_i|b_i')$
    end for
    if $s \geq B$ and $s\%E = 0$ then
            $A \leftarrow (\theta^s, a^s, b^s)$
    end if
end for
return average of A
```

To estimate the window size parameter L_t that maximizes the response prediction accuracy, we propose the use of the greedy algorithm. The greedy algorithm is a problem-solving paradigm where the local optimum choice is made in each stage with the expectation of finding the global optimum solution. The optimal variance parameter δ_t is obtained by calculating the prediction for $\delta = \{0.1, \dots, 1.0\}$, then taking the variance with the maximum prediction accuracy. The process of estimating the window size parameter L_t and variance parameter δ_t with the greedy algorithm is as follows:

1. For each item, calculate the response prediction accuracy with the initial values of the window size parameter L_t and variance parameter δ_t.
2. Calculate and compare the response prediction accuracies with the window sizes $L_t - 1$ and $L_t + 1$.
3. Select the best of the three prediction accuracy rates, then repeat step 2 by increasing or decreasing the window size L_t depending on the selection; if the prediction accuracy with the window size $L_t - 1$ is better, decrease the window size L_t by 1, if the window size $L_t + 1$ one is better, increase the window size L_t by 1. Repeat until the prediction accuracy calculated using L_t is better than the accuracy rates calculated using $L_t - 1$ and $L_t + 1$.
4. The variance parameter δ_t is obtained by calculating the prediction for $\delta =$

$\{0.1, \ldots, 1.0\}$, then taking the variance with the maximum prediction accuracy.

5. Take the window size parameter and the variance parameter that result in the best prediction accuracy for the current item and set them as the initial values for the next item.
6. Repeat step 1 for the next item.

A flowchart of the estimation of the window size and variance parameters is shown in Fig. 4.

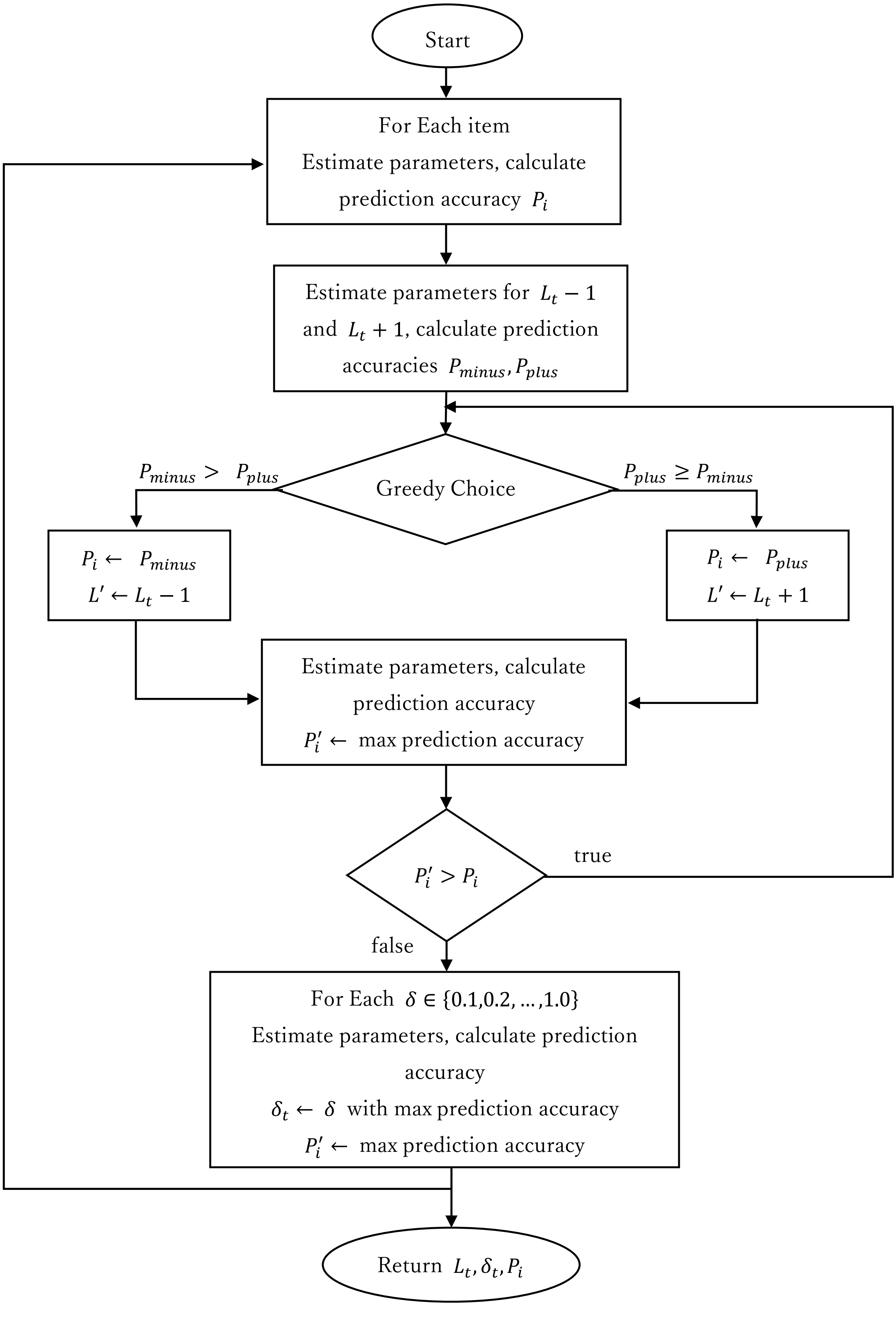

Fig 4 Flowchart of window size and variance parameter estimation

6. Experiment

To evaluate the estimates of learner’s ability produced by the proposed model, the learner’s ability parameter was estimated, then used to predict the learner's response. After obtaining the predicted response, the prediction accuracy was calculated using the real test data, and the results were compared with those of the HMIRT model and traditional IRT model. The data used in this study consisted of a number of learning tasks within three courses:

(1) Foundation of programming 1 (7 tasks, 148 learners)

(2) Foundation of programming 2 (18 tasks, 75 learners)

(3) Information Society and Information Ethics (13 tasks, 23 learners)

These data are taken from the SAMURAI e-learning system for university students (Ueno,2004) [14]. We performed 10-fold cross-validation in the experiment to reduce over-fitting and generalize the prediction accuracy.

6.1. Response Prediction Accuracy

After obtaining the learner’s ability, the response for each item can be predicted by calculating the probability of the learning getting the correct answer using equation (1) and then setting the response as follows:

$$\text{Predicted response} \begin{cases} \text{0: incorrect if the probability is less than 0.5} \\ \text{1: correct if the probability is more than 0.5} \end{cases}$$

After obtaining the predicted response for each item, it is checked against the real response data, and overall prediction accuracy is calculated by taking the average accuracy of all of the items. Here, the first item’s response will not be used to calculate the average prediction due to the fact that the learner must first undertake the first task before the system can use their response for the later tasks.

Table 1 shows that the response prediction accuracy of the proposed model is better than those of both the HMIRT model and the traditional IRT model for all three datasets. Figures 5–7 show the graphs of the prediction accuracies of all three models for each item in each of the three datasets. From these graphs, we can see that the predictions for the earlier time states tend to be the same for all three models, especially for a small dataset, but the model predictions gradually diverge as the learning progresses. For the Foundation of Programming 1 dataset (Fig. 5), the prediction accuracy of the proposed model is slightly better than those of the other models for item 2 and exactly the same as the other models for item 3. From item 4 onward, the proposed model clearly performs better than the IRT model and slightly better than HMIRT. For the Foundation of Programming 2 dataset (Fig. 6), due to the large size of the dataset, the prediction accuracy of the proposed model is clearly better from the beginning than both the HMIRT model and IRT model. On the other hand, for the smaller Information Society and Information Ethics dataset (Fig. 7), the prediction accuracy of the proposed model is exactly the same as the other two models from the beginning until item 8. Beginning at item 9, the prediction accuracies of the proposed model and HMIRT are better than that of the IRT model, and from item 11 onward, the proposed model performs better than HMIRT.

Table 1: Average prediction accuracy.

Dataset	Proposed Model	HMIRT	IRT
Foundation of programming 1	**78.30%**	75.26%	69.84%
Foundation of programming 2	**81.69%**	76.17%	71.26%
Information Society and Information Ethics	**90.00%**	87.91%	85.00%

Table 2: F-measurement.

Dataset		AFHMIRT	HMIRT	IRT
Foundation of programming 1	F-Measure of correct response	**0.823504**	0.7929848	0.7728592
	F-Measure of incorrect response	**0.725104**	0.6462029	0.5614179
	Avareage F-Measure	**0.774304**	0.71959388	0.667138548
Foundation of programming 2	F-Measure of correct response	**0.853826567**	0.82707805	0.78244226
	F-Measure of incorrect response	**0.636743631**	0.607326504	0.484960642
	Avareage F-Measure	**0.745285099**	0.717202277	0.633701451
Information Society and Information Ethics	F-Measure of correct response	**0.960675274**	0.926978679	0.914302032
	F-Measure of incorrect response	**0.501904762**	0.282857143	0.116450217
	Avareage F-Measure	**0.731290018**	0.604917911	0.515376124

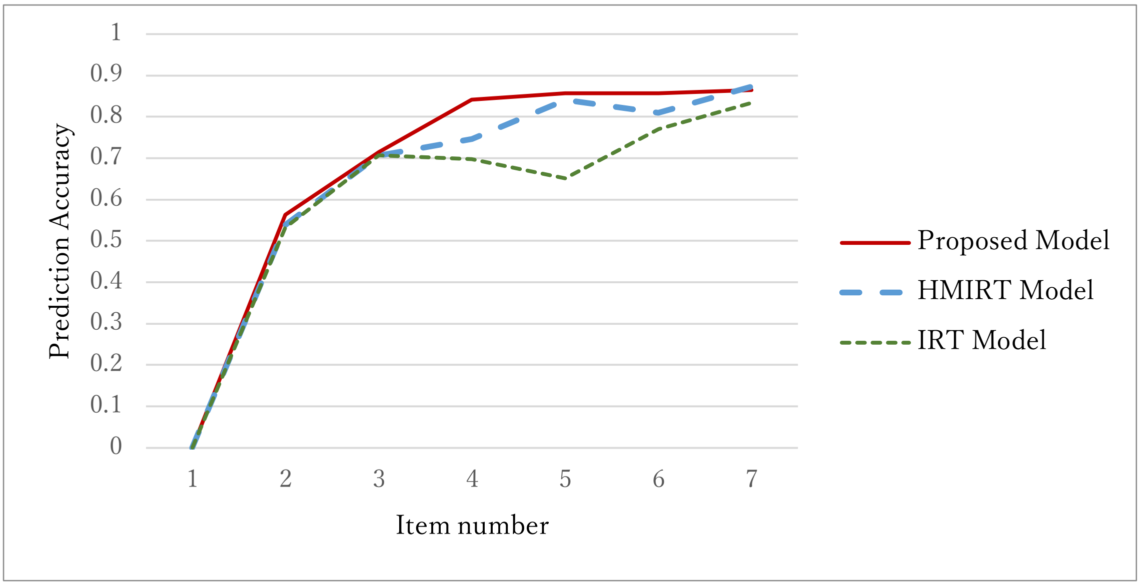

Fig. 5 Prediction accuracy of Foundation of Programming 1 (7 tasks, 148 learners).

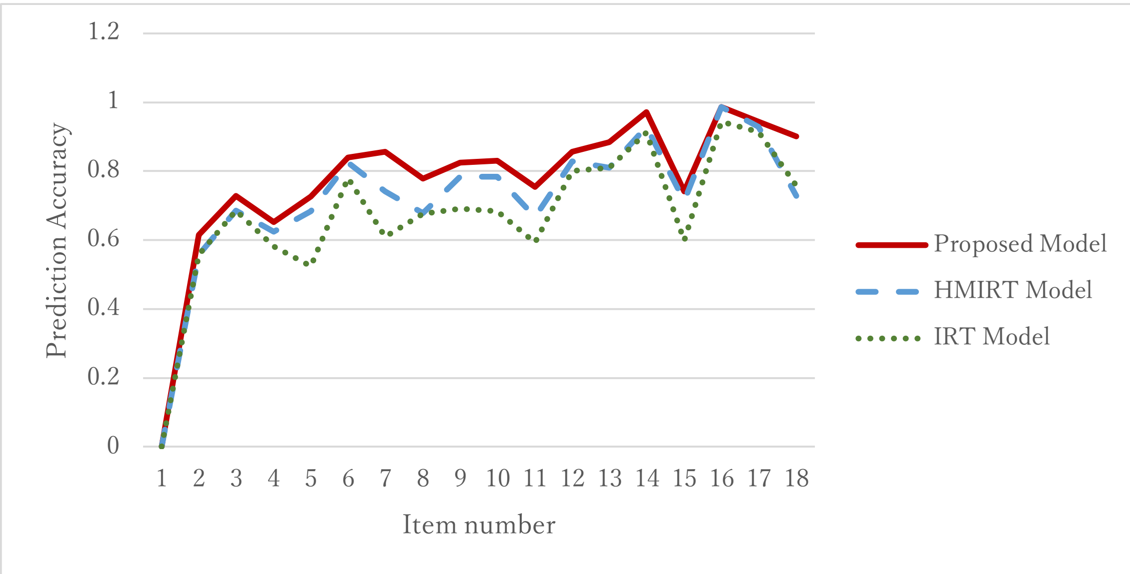

Fig. 6 Prediction accuracy of Foundation of Programming 2 (18 tasks, 77 learners).

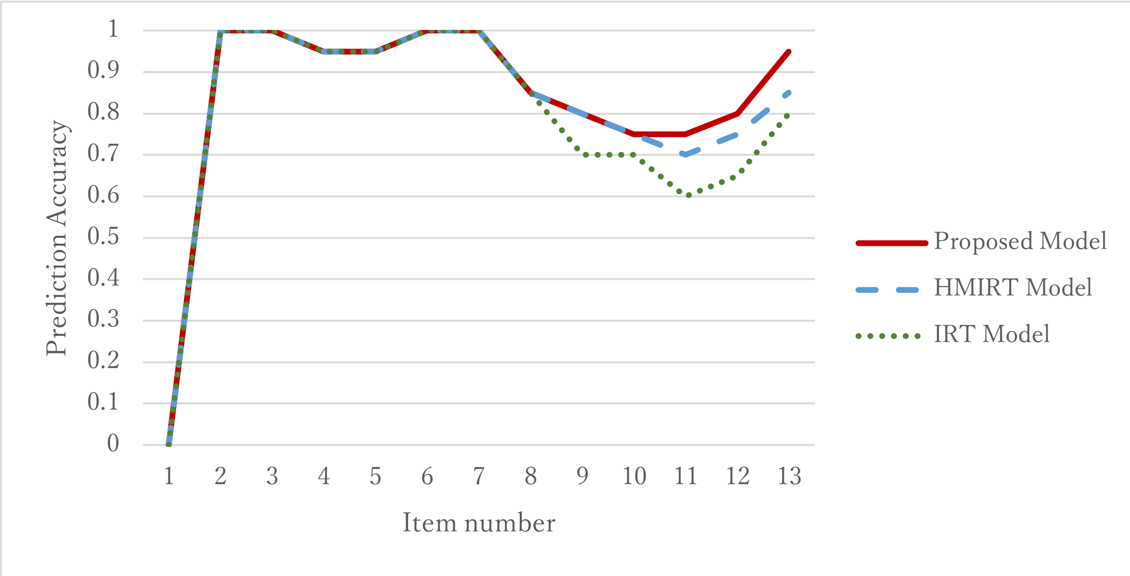

Fig. 7 Prediction accuracy of Information Society and Information Ethics (13 tasks, 23 learners).

6.2. Window Size Parameter

Figures 8–10 show how the window size changed during the learning process. Fig. 8 shows that for the Foundation of Programming 1 dataset, the window size tended to change only a small amount in the early time states, with larger changes later on. This can be related to the prediction accuracy in Fig. 5, where the prediction accuracy of the proposed model only changes slightly compared with the prediction accuracy of the HMIRT model in the first 3 tasks. Fig. 9 clearly shows the changes in the window size parameter for each item in the Foundation of Programming 2 dataset. The prediction accuracy of this dataset (Fig. 6) also shows that the proposed model has a better prediction accuracy. However, in Fig. 10, where the size of the Information Society and Information Ethics dataset is small, the window size does not change for any time state.

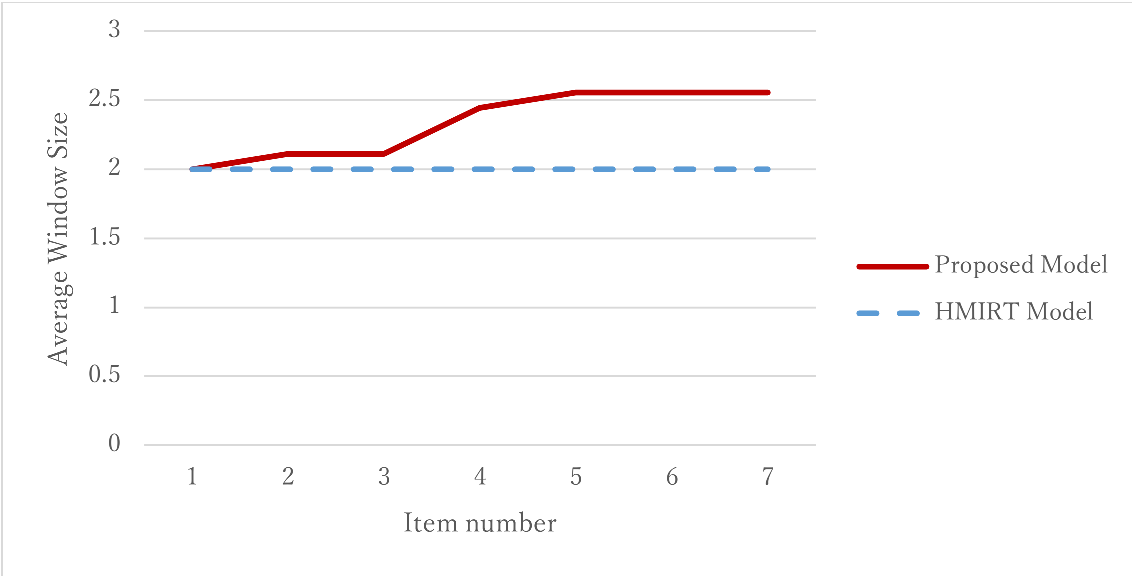

Fig. 8 Window size of Foundation of Programming 1
(7 tasks, 148 learners).

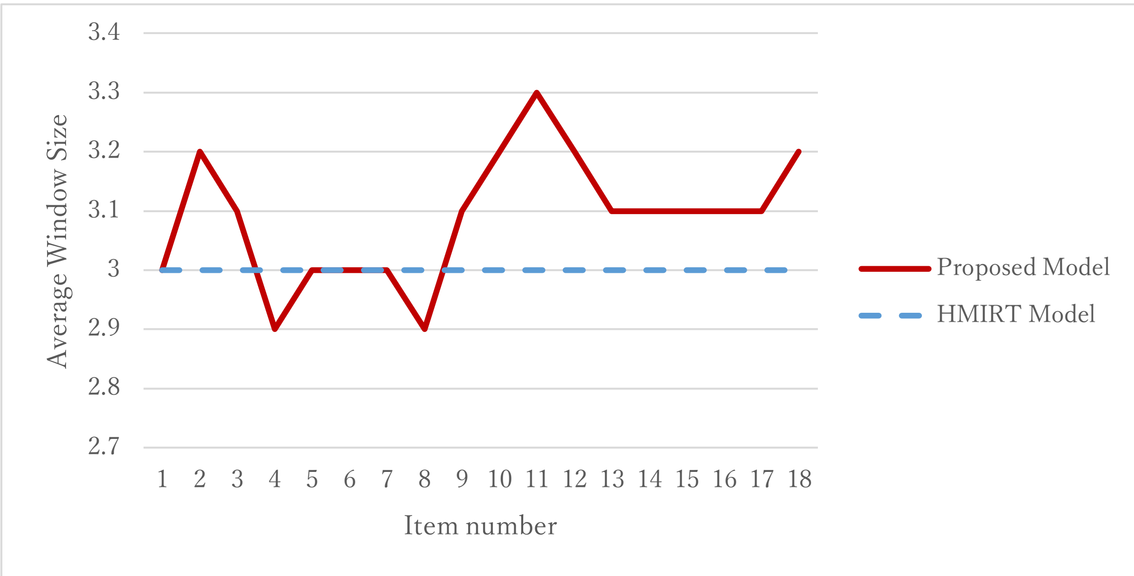

Fig. 9 Window size of Foundation of Programming 2
(18 tasks, 77 learners).

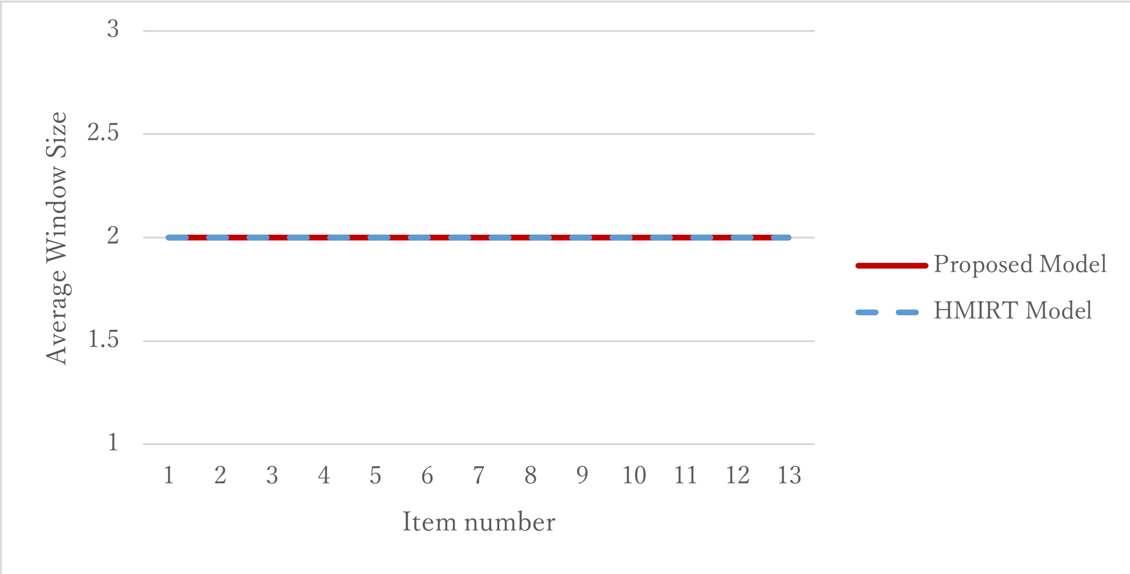

Fig. 10 Window size of Information Society and Information Ethics (13 tasks, 23 learners).

6.3. Variance Parameter

Figures 11–13 show how the variance parameter was set for each item. Fig. 11 shows that the variance begins high, then gradually decreases. This means that for the Foundation of Programming 1 dataset, the learner's ability will likely change by some large amount at first, then as the learning progresses, the changes in the learner's ability will be smaller. In Fig. 12, showing the Foundation of Programming 2 dataset, the variance of the proposed model starts off quite low, then increases as the learning progresses. The variance peaks at item 11, then starts to fall until the end of the learning process. In Fig. 13, representing the Information Society and Information Ethics dataset where the dataset size is small, the variance of the proposed model is exactly the same as that of the HMIRT model from the beginning to item 10. This can be related to the predictions of this dataset (Fig. 7), as the predictions of the proposed model are exactly the same as those of the

HMIRT model from the beginning until item 10. However, from item 11 onward, by decreasing the variance, the prediction accuracy of the proposed model is now better than that of HMIRT.

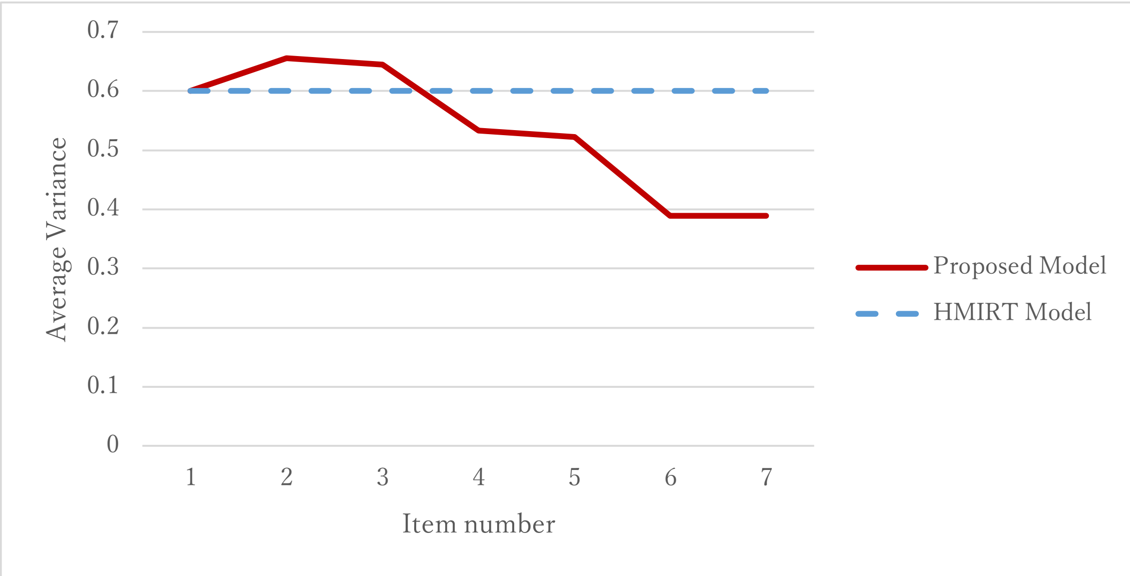

Fig. 11 Variance of Foundation of Programming 1
(7 tasks, 148 learners).

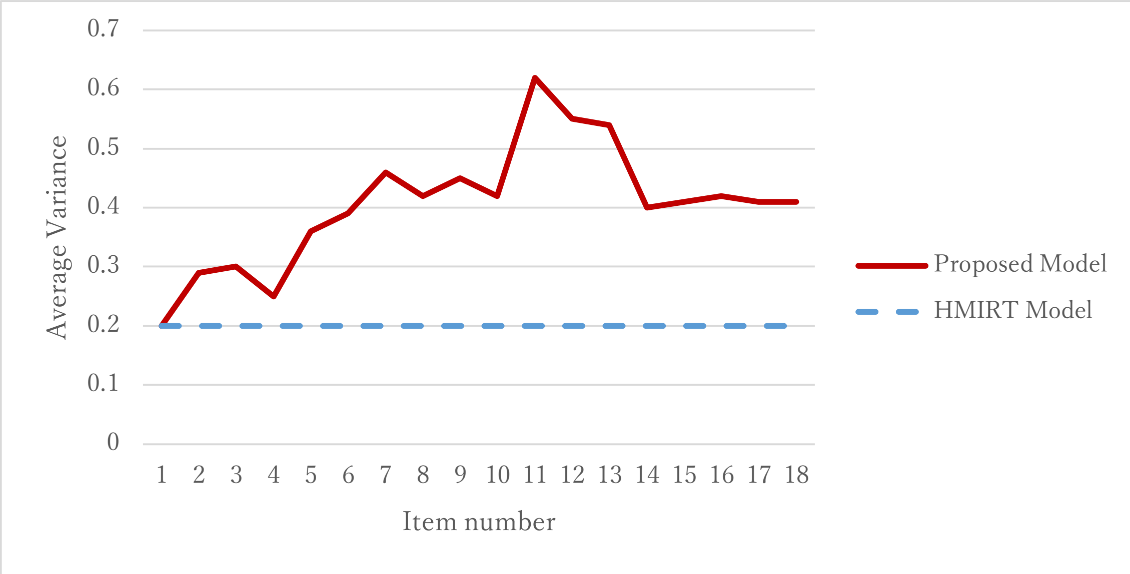

Fig. 12 Variance of Foundation of Programming 2
(18 tasks, 77 learners).

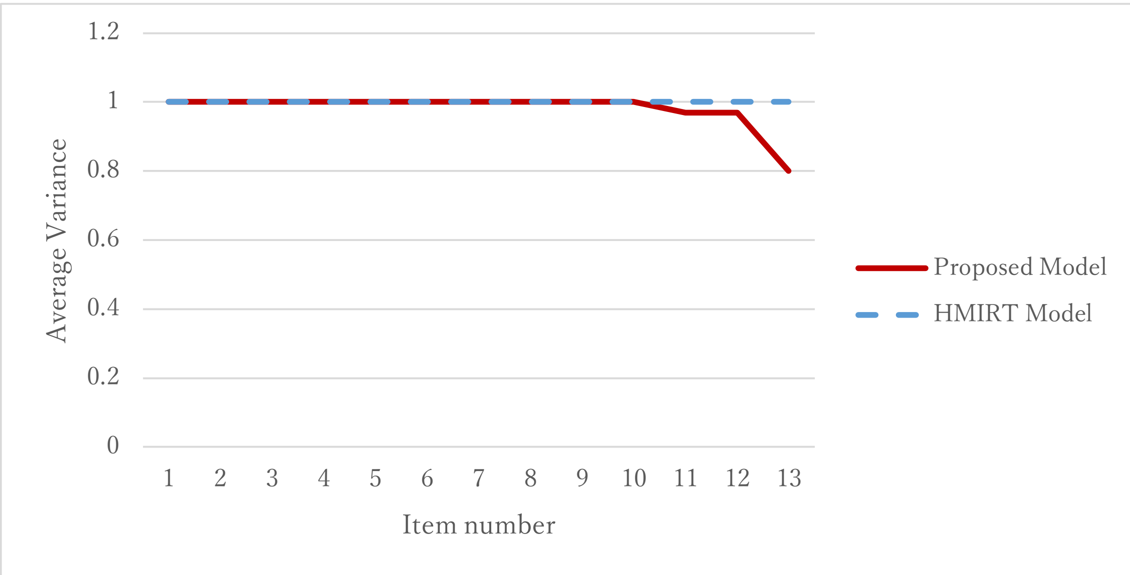

Fig. 13 Variance of Information Society and Information
Ethics (13 tasks, 23 learners).

7. Conclusion

In this research, we proposed a new method to estimate the learner's ability from the learning data, then used the estimated ability to predict the response for future tasks. The proposed model, AFHMIRT, generalizes the Hidden Markov Item Response Theory and replaces the fixed values of the window size and variance parameters with time-series so that the parameters can fluctuate as learning progresses. In addition, we also proposed using a greedy algorithm to estimate the window size parameter. From the results of the experiment, we demonstrated that modeling the window size and variance parameters as time-series rather than fixed values resulted in a better prediction accuracy. Moreover, the responses were predicted by the proposed model for one item at a time, whereas the HMIRT model predicts the responses for all items at once. This made the proposed model's predictions more precise. However, the proposed model has a disadvantage with respect to estimation time. As described in Section 4, the proposed model needs to re-estimate the item parameter every time the window size or the variance changes, which requires a lot of time to run, especially for larger datasets. Improving the estimation time will be considered in future tasks.

8. Reference

[1] L.S. Vygotsky, Thought and language,Harvard University Press,1962.

[2] L.S. Vygotsky, Mind in society,Harvard University Press,1978.

[3] D.Wood, J.S.Bruner, and G.Ross, ” The role of tutoring in problem solving.” ,Journal of child psychiatry and psychology, and allied disciplines, pp.89-100,1976.

[4] A.Collins, "JS & Newman,SE(1989).Cognitive apprenticeship: teaching the craft of reading, writing and mathematics,” Resnick, LB Knowing, learning and instruction, pp.453–494, 1989.

[5] J.Bruner, The Culture of Education, Harvard University Press, 1996, 1996.

[6] 植野真臣, 松尾淳哉, "項目反応理論を用いて適応的ヒントを提示する足場かけシステム", 電子情報通信学会論文誌 D, Vol.J98-D,No.1,pp.17-29,Jan.2015.

[7] M. Ueno, and Y. Miyazawa, "Probability based scaffolding system with fading", Artificial Intelligencein Education - 17th International Conference, AIED 2015. pp.492-503,2015.

[8] M. Ueno, Y. Miyazawa: IRT-Based Adaptive Hints to Scaffold Learning in Programming, IEEE Transactions on Learning Technologies, IEEE computer Society, Vol.11, Issue 4, 415-428 (2018)

[9] Baker,B,F, and Kim,S.: Item Response Theory: Parameter Estimation Techniques, Second Edition, NY:Marcel Dekker, Inc,2004(2004)

[10] E. Tsutsumi, M. Uto， M. Ueno: "Bayesian Knowledge Tracing の一般化としての隠れマルコフ IRT モデル"，人工知能学会 2019 年度人工知能学会全国大会（第 33 回),2019

[11] 堤 瑛美子, 宇都 雅輝, 植野 真臣. ”時系列学習データを用いた隠れマルコフ IRT による高精度パフォーマンス予測”,日本行動計量学会 第 47 回大会, 2019

[12] F.M. Lord, Applications of item response theory to practical testing problems,

Mahwah, NJ: Lawrence Erlbaum Associates, Inc,1980.

[13] S.Impedovo, A. Ferrante and R Modugno,"HMM Based Handwritten Word Recognition System by Using Singularities," Document Analysis and Recognition, 2009. ICDAR ’09. 10th International Conference on,2009.

[14] J.Ortiz, A.G.Olaya and D. Borrajo,"A Dynamic Sliding Window Approach for Activity Recognition"UMAP’11 Proceedings of the 19th international conference on User modeling, adaption, and personalization,pp.219-230,2011.

[15] M. Ueno, "Data mining and text mining technologies for collaborative learning in an ILMS Samurai", Proc. IEEE Int. Conf. Adv. Learning Technol., pp. 1052-1053, 2004.