Supplemental Material for "Causal-Driven Skill Prerequisite Structure Discovery"

Anonymous submission

Parameter Learning

We specify the EM algorithm for the LGLVM model, of which the parameters include $\{\boldsymbol{W}, \boldsymbol{\Phi}, \boldsymbol{\Theta}\}$, where $\boldsymbol{W} = [\boldsymbol{W}_{nk}]$, $\boldsymbol{\Phi} = [\phi_1, \phi_2, \cdots, \phi_{\mathrm{N}}]^{\top}$, and $\boldsymbol{\Theta}$ is a diagonal matrix with diagonal entries $\{\theta_n\}$ $(1 \leq n \leq \mathrm{N})$. Learning the parameters of the proposed model amounts to estimating the parameters for the regression model

$$\mathbf{O}_n | \mathbf{Z} \sim \mathcal{N}\left(\sum_{k=1}^{\mathrm{K}} \mathbf{Q}_{nk} \boldsymbol{W}_{nk} Z_k + \phi_n, \theta_n\right).$$

First of all, we note the joint probability distribution over $(\mathbf{O}, \mathbf{Z})$ can be expressed as

$$P(\mathbf{O}, \mathbf{Z}) = P(\mathbf{O} \mid \mathbf{Z})\, P(\mathbf{Z}). \tag{1}$$

As our goal is to discover the correlation relationships among skills from all students' performance data, we ignore the individual differences among students and use each student's performance on a particular exercise as a repeated observation of it. Hence, given a set of repeated samples $\{\mathbf{o}^1, \mathbf{o}^2, \cdots, \mathbf{o}^d, \cdots, \mathbf{o}^{\mathrm{D}}\}$, where $\mathbf{o}^d = \{o_1^d, o_2^d, \cdots, o_n^d, \cdots, o_{\mathrm{N}}^d\}$, we assume that these observations are mutually independent and identically distributed (i.i.d.), then the expression in the right-hand side of Eq. (1) is specified as follows

$$P(\mathbf{O} \mid \mathbf{Z}) = \prod_{d=1}^{\mathrm{D}} \prod_{n \in \mathcal{I}(d)} \left[(2\pi\theta_n)^{-\frac{1}{2}} \times \exp\left\{ -\frac{1}{2\theta_n} \left(O_n^d - \left[\sum_{k=1}^{\mathrm{K}} \mathbf{Q}_{nk} \boldsymbol{W}_{nk} Z_k + \phi_n \right] \right)^2 \right\} \right],$$

$$P(\mathbf{Z}) = \mathcal{N}(\mathbf{Z} \mid \boldsymbol{\mu}_{\mathbf{Z}}, \mathbf{I}),$$

where $\mathcal{I}(d)$ is the set of exercises that the student St_d has done.

For the E-step of the EM algorithm, one needs to calculate the expected data-complete log-likelihood of the model, which gives

$$\begin{aligned}
\mathcal{Q} &= \mathbb{E}[\log P(\mathbf{O}, \mathbf{Z})] \\
&= -\frac{\sum_{d=1}^{\mathrm{D}} |\mathcal{I}(d)|}{2} \log 2\pi - \frac{1}{2} \sum_{d=1}^{\mathrm{D}} \sum_{n \in \mathcal{I}(d)} \log \theta_n \\
&\quad - \sum_{d=1}^{\mathrm{D}} \sum_{n \in \mathcal{I}(d)} \frac{1}{2\theta_n} \mathbb{E}\left[\left(o_n^d - \left[\sum_{k=1}^{\mathrm{K}} \mathbf{Q}_{nk} \boldsymbol{W}_{nk} Z_k + \phi_n \right] \right)^2 \right] \\
&\quad - \frac{\mathrm{K} \cdot \mathrm{T}}{2} \log 2\pi - \frac{1}{2} \mathbb{E}\left[\mathbf{Z}^{\top} \mathbf{Z} \right] + \boldsymbol{\mu}_{\mathbf{Z}}^{\top} \mathbb{E}[\mathbf{Z}] - \frac{1}{2} \boldsymbol{\mu}_{\mathbf{Z}}^{\top} \boldsymbol{\mu}_{\mathbf{Z}},
\end{aligned}$$

where $\mathbb{E}[\cdot] = \mathbb{E}_{\mathbf{Z} \sim g(\mathbf{Z}|\mathbf{O})}[\cdot]$ denotes the expectation of all the latent variables $\mathbf{Z}$ conditioned on the observations $\mathbf{O}$. The following M-step for the EM algorithm can now be derived by computing the partial derivatives of the expected log-likelihood function.

Specifically, for the parameter θ_n we now get

$$\frac{\partial \mathcal{Q}}{\partial \theta_n} = \sum_{d=1}^{\mathrm{D}} \delta_{dn} \frac{1}{2\theta_n^2} \left\{ \mathbb{E}\left[\left(o_n^d - \left[\sum_{k=1}^{\mathrm{K}} \mathbf{Q}_{nk} \boldsymbol{W}_{nk} Z_k + \phi_n \right] \right)^2 \right] - \theta_n \right\},$$

where $\delta_{dn} = 1$ if there exists an answer record of the student St_d on the exercise Ex_n. The updating rule for θ_n becomes

$$\hat{\theta}_n \leftarrow \frac{\sum_{d=1}^{\mathrm{D}} \delta_{dn} \mathbb{E}\left[\left(o_n^d - \left[\sum_{k=1}^{\mathrm{K}} \mathbf{Q}_{nk} \boldsymbol{W}_{nk} Z_k + \phi_n \right] \right)^2 \right]}{\sum_{d=1}^{\mathrm{D}} \delta_{dn}}, \tag{2}$$

and the expectation can be expanded as

$$\begin{aligned}
&\mathbb{E}\left[\left(o_n^d - \left[\sum_{k=1}^{\mathrm{K}} \mathbf{Q}_{nk} \boldsymbol{W}_{nk} Z_k + \phi_n \right] \right)^2 \right] \\
&= \left[o_n^d \right]^2 - 2 o_n^d \phi_n + \phi_n^2 + 2 \left(\phi_n - o_n^d \right) \sum_{k=1}^{\mathrm{K}} \mathbf{Q}_{nk} \boldsymbol{W}_{nk} \mathbb{E}[Z_k] \\
&\quad + \sum_{i=1}^{\mathrm{K}} \sum_{j=1}^{\mathrm{K}} \mathbf{Q}_{ni} \mathbf{Q}_{nj} \boldsymbol{W}_{ni} \mathbb{E}\left[Z_i Z_j^{\top} \right] \boldsymbol{W}_{nj},
\end{aligned}$$

which involves the expectations $\mathbb{E}[Z_k]$ and $\mathbb{E}\left[Z_i Z_j^{\top}\right]$.

Next, for the parameter ϕ_n we get

$$\frac{\partial \mathcal{Q}}{\partial \phi_n} = -\sum_{d=1}^{\mathrm{D}} \delta_{dn} \frac{1}{2\theta_n} \left(-2 o_n^d + 2 \sum_{k=1}^{\mathrm{K}} \mathbf{Q}_{nk} \boldsymbol{W}_{nk} \mathbb{E}[Z_k] + 2\phi_n \right),$$

and therefore

$$\hat{\phi}_n \leftarrow \frac{\sum_{d=1}^{\mathrm{D}} \delta_{dn} o_n^d - \sum_{d=1}^{\mathrm{D}} \delta_{dn} \sum_{k=1}^{\mathrm{K}} \mathbf{Q}_{nk} \boldsymbol{W}_{nk} \mathbb{E}[Z_k]}{\sum_{d=1}^{\mathrm{D}} \delta_{dn}}. \quad (3)$$

Finally, the updating rule for $\boldsymbol{W}_{nk}$ follows from

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{W}_{nk}} = -\sum_{d=1}^{\mathrm{D}} \delta_{dn} \frac{1}{2\theta_n} \Bigg(-2o_n^d \mathbf{Q}_{nk} \mathbb{E}[Z_k] + 2\phi_n \mathbf{Q}_{nk} \mathbb{E}[Z_k] + 2 \sum_{j=1, j\neq k}^{\mathrm{K}} \mathbf{Q}_{nk} \mathbf{Q}_{nj} \mathbb{E}\left[Z_k Z_j^\top\right] \boldsymbol{W}_{nj} + 2\mathbf{Q}_{nk} \mathbb{E}\left[Z_k Z_k^\top\right] \boldsymbol{W}_{nk} \Bigg),$$

and is given by

$$\hat{\boldsymbol{W}}_{nk} \leftarrow \left[\sum_{d=1}^{\mathrm{D}} \delta_{dn} \mathbf{Q}_{nk} \mathbb{E}\left[Z_k Z_k^\top\right]\right]^{-1} \times \left[\sum_{d=1}^{\mathrm{D}} \delta_{dn} \mathbf{Q}_{nk} \left(o_n^d - \phi_n\right) \mathbb{E}[Z_k] - \sum_{d=1}^{\mathrm{D}} \delta_{dn} \sum_{j=1, j\neq k}^{\mathrm{K}} \mathbf{Q}_{nk} \mathbf{Q}_{nj} \mathbb{E}\left[Z_k Z_j^\top\right] \boldsymbol{W}_{nj}\right]. \quad (4)$$

As can be seen, the updating rules of θ_n, ϕ_n, and $\boldsymbol{W}_{nk}$, for $1 \leq n \leq \mathrm{N}, 1 \leq k \leq \mathrm{K}$, involve the expectations $\mathbb{E}[Z_k]$ and $\mathbb{E}\left[Z_i Z_j^\top\right]$. As there may exist missing values in each student's answer records, given the d-th observation $\mathbf{o}^d$, we see that the joint distribution over the latent variables $\mathbf{Z}$ conditioned on $\mathbf{o}^d$ is also Gaussian, which can be calculated as follows

$$\mathbf{Z} \mid \mathbf{o}^d \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{Z}}^c, \boldsymbol{\Sigma}^c\right), \quad (5)$$

where the (conditional) mean vector and covariance matrix is given by

$$\boldsymbol{\mu}_{\mathbf{Z}}^c = \boldsymbol{\Sigma}^c \left[(\mathbf{Q} \odot \boldsymbol{W})^\top \boldsymbol{\Theta}^{-1} \left(\mathbf{o}^d - \tilde{\boldsymbol{\Phi}}\right) + \mathbf{I}^{-1} \boldsymbol{\mu}_{\mathbf{Z}}\right],$$
$$\boldsymbol{\Sigma}^c = \left[\mathbf{I}^{-1} + (\mathbf{Q} \odot \boldsymbol{W})^\top \boldsymbol{\Theta}^{-1} (\mathbf{Q} \odot \boldsymbol{W})\right]^{-1}.$$

In Eq. (5), $\boldsymbol{W}$ is a loading matrix, which is formed by $\boldsymbol{W}_{nk}$. $\boldsymbol{\Theta}$ is a diagonal matrix, where the n-th element on the diagonal is θ_n. Similarly, the parameters ϕ_n compose $\tilde{\boldsymbol{\Phi}}$. Note that we omit the n-th entry in $\tilde{\boldsymbol{\Phi}}$ if the corresponding value is missing in $\mathbf{o}^d$, i.e., the student has no answer record in the exercise Ex_n. Therefore, $\mathbb{E}[Z_k]$ can be extracted from the mean vector in Eq. (5), and $\mathbb{E}\left[Z_i Z_j^\top\right]$ is calculated as

$$\mathbb{E}\left[Z_i Z_j^\top\right] = \boldsymbol{\Sigma}_{i,j}^c + \mathbb{E}[Z_i] \mathbb{E}[Z_j]^\top,$$

where $\boldsymbol{\Sigma}_{i,j}^c$ is the sub-matrix of $\boldsymbol{\Sigma}^c$ restricted to Z_i and Z_j.

We summarize the LGLVM model training algorithm in Algorithm 1. We start with the parameter initialization, including initializing $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, and $\boldsymbol{W}$ (line 1). It then proceeds through multiple iterations (lines 2-10). In each loop, we update one parameter in turn by keeping the other parameters constant. For example, we update θ_n via Eq. (2) by fixing the parameters $\boldsymbol{\Phi}$ and $\boldsymbol{W}$ (line 4). Finally, we output all parameters till the algorithm converges (line 11).

Algorithm 1: Parameter Learning of the LGLVM Model

Input: The student response log $\mathbf{o}$, the Q-matrix $\mathbf{Q}$.
Output: $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, and $\boldsymbol{W}$.

1: Initialize $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, and $\boldsymbol{W}$
2: **while** *not converged* **do**
3: **for** $n = 1, 2, \cdots, \mathrm{N}$ **do**
4: Fix $\boldsymbol{\Phi}$ and $\boldsymbol{W}$, update θ_n by Eq. (2)
5: Fix $\boldsymbol{W}$, update ϕ_n by Eq. (3)
6: **for** $k = 1, 2, \cdots, \mathrm{K}$ **do**
7: Fix $\boldsymbol{\Phi}$ and $\boldsymbol{W}_{nj}$ $(1 \leq n \leq \mathrm{N}, 1 \leq j \leq \mathrm{K}$, and $j \neq k)$, update $\boldsymbol{W}_{nk}$ by Eq. (4)
8: **end for**
9: **end for**
10: **end while**
11: **return** $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, and $\boldsymbol{W}$

Time Complexity of the CSPS Model

In this section, we discuss the time complexity of our CSPS model, which is composed of two stages. In the first stage, CSPS invokes the LGLVM model to output the covariance of the latent skills via the updating rules, i.e., Eq. (2), Eq. (3), and Eq. (4) (see also Algorithm 1). According to the updated rules, it is not hard to compute the arithmetic operations. In each iteration, the time complexity is $O(\mathrm{DNK}^2)$ for the measurement noise $\boldsymbol{\Theta}$, $O(\mathrm{DNK})$ for the exercise average complexity $\boldsymbol{\Phi}$, and $O(\mathrm{DNK}^2)$ for the loading matrix $\boldsymbol{W}$. Thus, the total time complexity in the first stage is $O(\#iter \times (\mathrm{DNK}^2))$, where $\#iter$ is the number of iterations needed for convergence.

In the second stage, the CSPS model uses the two-phase framework to identify the skill prerequisite structure. In **Phase I**, given the target variable Z_k, the proposed NB-search algorithm first calculates the association value of every candidate with Z_k and then ranks all variables in order of the association values, thus, the time complexity is $O(\mathrm{K} \log \mathrm{K})$. Next, the NB-search tests for the (partial) independence of every variable that is added into $\mathbf{NB}(Z_k)$ with Z_k conditioned on all subsets of the remaining variables in $\mathbf{NB}(Z_k)$. In the worst case, the number of tests is bounded by $O(\mathrm{K} \times |\mathbf{NB}(\mathrm{Z_k})|^4)$. Hence, the time complexity in **Phase I** is bounded by $O(\mathrm{K}^2 \times |\mathbf{NB}|^4)$, where **NB** is the largest set of neighbors over all variables. In **Phase II**, the most time-consuming operations of the DS-SPV algorithm are to identify the potential Markov blanket for every variable, which consists of the target variable and its neighbors. However, we can reduce operations by pre-storing the neighbors of a target one when employing the NB-search algorithm in **Phase I**, thus, with caching all the calls to the NB-search the overall cost in **Phase II** is $O(\mathrm{K}^4)$ for calculating the SPV values for all variables in $\mathbf{Z}$. In summary, the total time complexity in the second stage is bounded by $O(\mathrm{K}^2 \times (|\mathbf{NB}|^4 + \mathrm{K}^2))$.

Data Processing of FrcSub and Alg0506

FrcSub. The real-world test data, i.e., FrcSub, is widely used in cognitive modeling (De La Torre 2009; DeCarlo 2011). The mathematical data set is made of binary test responses (right or wrong) of 535 examinees on 20 *Fraction-*

Table 1: Fraction subtraction data (partial) and the skill description

Exercise ID	Exercise	Skill IDs	Skill Description
Ex_1	$\frac{5}{3} - \frac{3}{4}$	$\mathrm{Sk}_4, \mathrm{Sk}_6, \mathrm{Sk}_7$	Sk_1: convert a whole number to a fraction
Ex_2	$\frac{3}{4} - \frac{3}{8}$	$\mathrm{Sk}_4, \mathrm{Sk}_7$	Sk_2: separate a whole number from a fraction
Ex_3	$\frac{5}{6} - \frac{1}{9}$	$\mathrm{Sk}_4, \mathrm{Sk}_7$	Sk_3: simplify before subtracting
Ex_4	$3\frac{1}{2} - 2\frac{3}{2}$	$\mathrm{Sk}_2, \mathrm{Sk}_3, \mathrm{Sk}_5, \mathrm{Sk}_7$	Sk_4: find a common denominator
Ex_5	$4\frac{3}{5} - 3\frac{4}{10}$	$\mathrm{Sk}_2, \mathrm{Sk}_4, \mathrm{Sk}_7, \mathrm{Sk}_8$	Sk_5: borrow from whole number part
Ex_6	$\frac{6}{7} - \frac{4}{7}$	Sk_7	Sk_6: column borrow to subtract the second numerator from the first
Ex_7	$3 - 2\frac{1}{5}$	$\mathrm{Sk}_1, \mathrm{Sk}_2, \mathrm{Sk}_7$	Sk_7: subtract numerators
...	...	...	Sk_8: reduce answers to the simplest form

Table 2: The brief description of the Alg0506 data set

Student ID	Exercise Name	Step Name	...	Correct First Attempt[1]	Incorrects	Hints[2]	Corrects	KC[3]	...
0BrbPbwCMz	EG4-FIXED	$3(x+2) = 15$	...	0	2	3	1	Eliminate Parens	...
0BrbPbwCMz	EG4-FIXED	$x + 2 = 15$	...	1	0	0	1	Remove Constant, Isolate Positive	...
0BrbPbwCMz	EG40	$4y = -10$	...	1	0	0	1	Remove Coefficient	...
0BrbPbwCMz	EG40	$4y/4 = -10/4$	...	1	0	0	1	Multiply/Divide	...
...	...	...	...	...	...	...	...	...	...

[1] Correct First Attempt denotes the tutor's evaluation of the student's first attempt on the step (1 is correct, 0 if an error).
[2] Hints records the total number of the requested hints by the student on the step.
[3] KC identifies skills that are used in an exercise, and a step can have multiple skills assigned to it.

Subtraction exercises measuring 8 skills. For a better illustration, Table 1 shows partial exercises coupled with the required skills from the fraction subtraction test. Each exercise is related to a different number of skills. For an exercise, there exists a prerequisite relationship among the involved skills. For example, if a student is required to solve the exercise Ex_3, he or she should know how to find a common denominator of $\frac{5}{6}$ and $\frac{1}{9}$ (Sk_4), before trying to subtract the numerators (Sk_7). Thus, Sk_4 can be considered as a prerequisite of SK_7.

Since FrcSub does not provide the true prerequisite structure, we invite five experts in the education area who are familiar with the *adding and subtracting fractions* to build the structure based on their logicality. The five volunteers are asked to reach agreements in manually labeling the prerequisite relationships among the required skills by working out all exercises in Table 1, which forms the sub-structures of the skills. For example, we obtain the sub-structure $\mathrm{Sk}_4 \rightarrow \mathrm{Sk}_7$ since all experts agree that Sk_4 is a prerequisite to Sk_7 when doing exercise Ex_2.

Alg0506. The public real-world log data of the 2005-2006 curriculum "Algebra I" [1], abbreviated as Alg0506, is collected from the interactions between students and computer-aided tutoring systems. The original data set incorporates the $813,661$ response records by 575 students. We give a brief description of the data set in Table 2, which includes student IDs, exercise names, and the involved skills (KC), as well as the answer records, such as the total number of incorrect attempts by a student on a step (Incorrects). As shown in Table 2, the Alg0506 data set outlines the exact steps for troubleshooting the given exercise (e.g., $3(x+2) = 15$ and $x + 2 = 15$ in EG4-FIXED), and the whole collection of steps for an exercise comprises the solution.

Before applying the proposed model and other comparative methods on Alg0506, we first preprocess the original data:

- (a) We filter out the exercises with less than 35 response records to guarantee that each exercise has enough observed data;
- (b) Since our goal is to discover the skills' prerequisite structure, as long as there is one step that lacks of skill descriptions, we delete the corresponding exercise; similarly, we remove the skills with unclear descriptions;
- (c) An exercise comprises several steps, each of which records a student's attempt result (e.g., correct or the number of incorrect attempts); to obtain the final response of a given student for a given exercise, we here define the *correct ratio* (denoted r) of the student's response to the exercise, i.e., $r = \mathrm{Num_c}/(\mathrm{Num_c} + \mathrm{Num_w})$, where $\mathrm{Num_c}$ and $\mathrm{Num_w}$ are the total numbers of correct and incorrect attempts of all steps for the exercise, respectively, and all the students' responses constitute the student response log.

After preprocessing, 438 students, 185 exercises, and 12 skills are included, each skill is tested on 50 exercises on average, and each exercise contains an average of six skills. Besides, the data density of the student response log is 21.05%, which means that there are approximately 80% of the exercises that a student has never done.

Since Alg0506 also lacks the true prerequisite structure, similar to the processing approach for FrcSub, given an exer-

[1] https://pslcdatashop.web.cmu.edu/KDDCup/

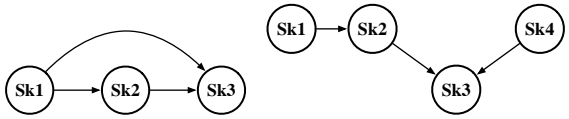
cise covering multiple steps, experts are asked to manually construct the prerequisite relationships step by step, which forms the local structure of the skills. To ensure the correctness of the output prerequisite relations, we omit the responses that are not answered correctly. After repeating the process, the overall prerequisite structure is generated by combining all local structures.

Additional Experiments

In this section, we conduct a series of additional experiments, including (1) the simulation study in the continuous data sets; (2) the performance evaluation in the binary real-world log data; (3) data type comparison; (4) prerequisite structure comparison; and (5) parameter sensitivity analysis.

Simulated Continuous Testing Data

We first compare the recovery quality for skills prerequisite structures of our method with CITS in the simulated continuous data sets, which are generated from the pre-designed prerequisite structure in Fig. 1. The experimental results in terms of the F_1-AR and F_1-OR (± two standard errors) are shown in Table 3. We can see that CSPS performs well in discovering adjacencies, especially recovering 100% of adjacencies for Sync1 when the sample size is greater than 150. Besides, our CSPS attains good performance for edge orientations in most cases. The results show that comparing CITS, which is also based on casual structure discovery algorithms, the proposed model possesses a satisfying ability to discover skills prerequisite structures.



(a) Sync1 (b) Sync2

Figure 1: Two structures that are used to generate the simulation data. Skills are represented as circles, and solid black arrows are the prerequisite relationships among skills. Exercise nodes are omitted for conciseness.

Performance Evaluation in the Binary Alg0506

We proceed to discuss the performance of the proposed method compared with baselines in a binary (correct/incorrect) copy of Alg0506, where for the observation o_n^d of the student St_d on the exercise Ex_n, the binary projection of that observation is set to zero if $o_n^d < \mathrm{mean}(\mathbf{o}_n)$, and one otherwise, to avoid choosing the cutoff points.

Fig. 2 shows the evaluation performance. Note that we do not give the result of the CMPD model because of its computational complexity, which makes learning prohibitive for data sets with a large number of skills. For the adjacency score in Fig. 2a, we can see that except for EPS-ND, CSPS achieves the best results (see F_1-AR), which shows that CSPS performs well in discovering adjacencies; while as observed in Fig. 2b, CSPS does not have a competitive performance

Table 3: Comparison of CSPS with CITS for discovering the prerequisite relationships in simulated continuous data sets

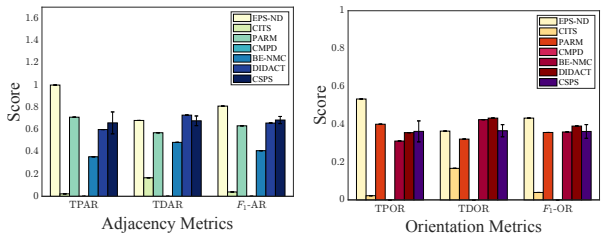
(a) Sync1

Sample Size	Metric	Model	
		CITS	CSPS
150	F_1-AR	0.400±0.000	0.940±0.096
	F_1-OR	0.400±0.000	0.587±0.128
500	F_1-AR	0.667±0.000	1.000±0.000
	F_1-OR	0.667±0.000	0.700±0.105
1000	F_1-AR	0.667±0.000	1.000±0.000
	F_1-OR	0.667±0.000	0.700±0.105
2000	F_1-AR	0.800±0.000	1.000±0.000
	F_1-OR	0.800±0.000	0.667±0.000

(b) Sync2

Sample Size	Metric	Model	
		CITS	CSPS
150	F_1-AR	0.800±0.000	0.770±0.116
	F_1-OR	0.800±0.000	0.606±0.167
500	F_1-AR	0.857±0.000	0.781±0.036
	F_1-OR	0.571±0.000	0.656±0.165
1000	F_1-AR	0.857±0.000	0.810±0.094
	F_1-OR	0.571±0.000	0.726±0.141
2000	F_1-AR	0.857±0.000	0.874±0.123
	F_1-OR	0.571±0.000	0.655±0.066

for edge orientations. A possible explanation is that the discretization of the observations compromises the identification ability of the proposed model.



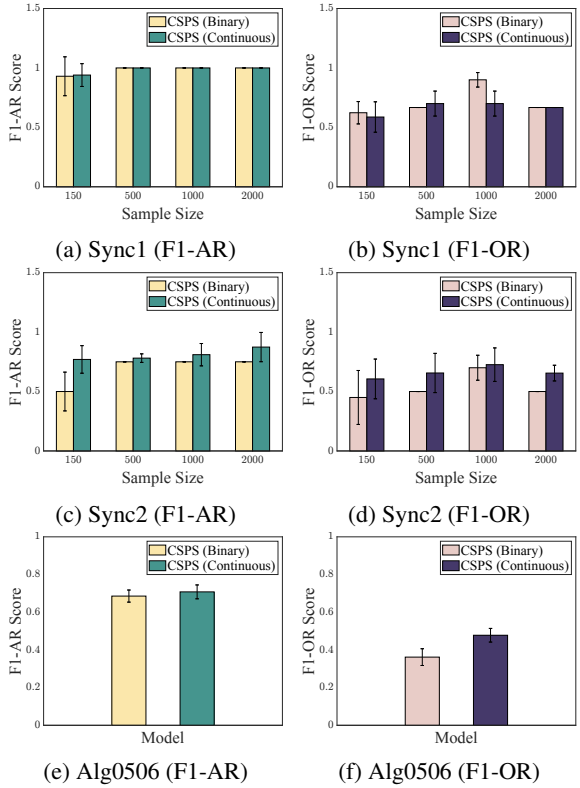
(a) Adjacency score (b) Orientation score

Figure 2: Comparison of CSPS with baselines for discovering the prerequisite relationships in binary Alg0506. Note that we do not give the results of CMPD because of its computational complexity (i.e., the execution time of the program is greater than a week).

Data Type Comparison: Binary vs. Continuous

Next, we compare the values of F_1-AR and F_1-OR for CSPS in continuous and binary data sets (i.e., Sync1, Sync2, and Alg0506), of which the results are shown in Fig. 3. We observe that the numerical results in continuous data sets are better than those in binary data sets, especially for Sync2 in terms of F_1-AR and F_1-OR and Alg0506 in terms of F_1-OR, which suggests that the continuous observations representing the partially correct responses of students on exercises

include more abundant information for inferring the prerequisite relations than binary observations.



(a) Sync1 (F1-AR) (b) Sync1 (F1-OR)

(c) Sync2 (F1-AR) (d) Sync2 (F1-OR)

(e) Alg0506 (F1-AR) (f) Alg0506 (F1-OR)

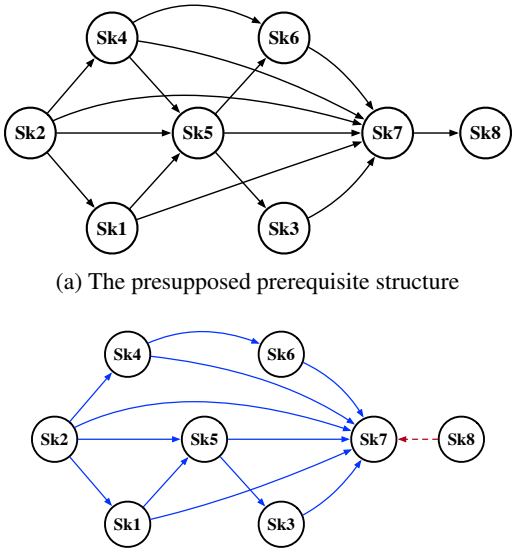
Figure 3: Comparison of CSPS in continuous and binary version of data sets.

Prerequisite Structure Comparison: Data-Driven Method vs. Expert Opinion

In this section, we compare the discovered prerequisite structure of skills by the proposed CSPS model with the presupposed one by the expert domain knowledge. To facilitate our description, we discuss the compared result in the FrcSub data set. Fig. 4 shows the details, where the solid blue arrows and red dashed arrows in Fig. 4b are used to denote the discovered results that agree and disagree with the presupposed structure shown in Fig. 4a, respectively. From Fig. 4, we have the following observations.

First, most of the prerequisite relationships discovered by our method follow the presupposed structure, which shows the effectiveness of the CSPS model in the discovery of skill prerequisite structure.

Second, in Fig. 4b, the presupposed prerequisites $\text{Sk}_4 \rightarrow \text{Sk}_5$ and $\text{Sk}_5 \rightarrow \text{Sk}_6$ are missing, and the prerequisite between Sk_7 and Sk_8 is opposite with the presupposed one. Besides recognizing the mistake of prerequisites of $\text{Sk}_4 \rightarrow \text{Sk}_5$ and $\text{Sk}_7 \rightarrow \text{Sk}_8$, for $\text{Sk}_5 \rightarrow \text{Sk}_6$, it can be seen that the data-driven method considers that the relationship between the two types of borrowing skills is full of obscure. According to the skill description in Table 1, an interesting interpretation is that if you have learned one of the skills, you can infer the other one. For example, knowing how to borrow from the hundreds column for $386-94$ can help to solve the exercise $3\frac{8}{4}-\frac{9}{4}$, and vice versa. Hence, the relationship can easily be overlooked or underestimated by domain experts due to its non-intuitive structure.



(a) The presupposed prerequisite structure

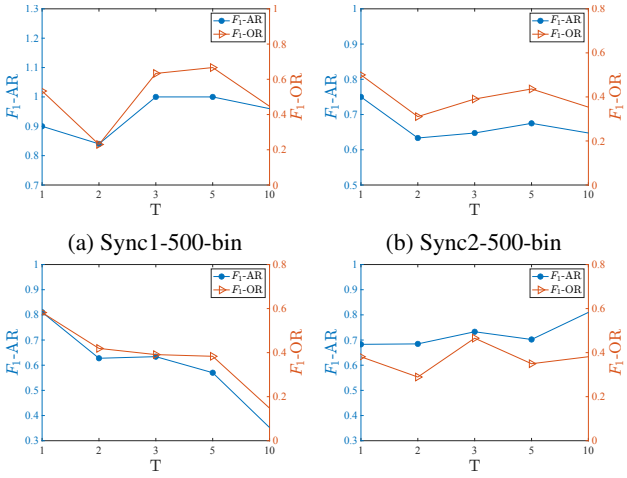
(b) The discovered prerequisite structure

Figure 4: Comparison of the prerequisite structure in FrcSub. Circles represent particular skills, and black arrows denote the prerequisite relationships among skills. We omit the skill-exercise mapping for conciseness.

Parameter Sensitivity Analysis

There is one tunable parameter in the CSPS model, i.e., the number of dimensions T for latent skills. Since it is hard to determine an appropriate value for T, we here use the grid search to find the best parameter. Specifically, we tune the value of T in the set $\{1, 2, 3, 5, 10\}$ for each data set, and analyze the parameter sensitivity in terms of the F_1-AR and F_1-OR. Note that in all the synthetic data sets, we only analyze the binary data with 500 observations for each structure, denoted as Sync1-500-bin and Sync2-500-bin, respectively, and the selected parameters are applied to the corresponding synthetic data sets with different sample sizes.

Fig. 5 visualizes the performance of varying the value of T in all data sets. It can be seen from Fig. 5a and Fig. 5b that as T increases, the value of F_1-AR and F_1-OR firstly decreases but increases when T surpasses 2, and finally drops at T = 10. Therefore, we set T = 5, 1 in Sync1-500-bin and Sync2-500-bin to obtain the best results. While for FrcSub in Fig. 5c, increasing the value of T leads to degradation in the performance, thus we choose T = 1 as the tuning result. For the Alg0506 data set, we observe from Fig. 5d that with T increase, the value of F_1-AR rises slowly, while the F_1-OR shows a trend of fluctuation, which reaches the peak when T = 3. Hence, we set T = 3 for the trading of the balance between the adjacency and orientation scores.



(a) Sync1-500-bin (b) Sync2-500-bin

(c) FrcSub (d) Alg0506

Figure 5: Sensitivity analysis of T on the four data sets.

References

De La Torre, J. 2009. DINA model and parameter estimation: a didactic. *Journal of Educational and Behavioral Statistics*, 34(1): 115–130.

DeCarlo, L. T. 2011. On the analysis of fraction subtraction data: the DINA model, classification, latent class sizes, and the Q-matrix. *Applied Psychological Measurement*, 35(1): 8–26.