

# Causal-Driven Skill Prerequisite Structure Discovery

Shenbao Yu<sup>1</sup>, Yifeng Zeng<sup>2\*</sup>, Fan Yang<sup>1\*</sup>, Yinghui Pan<sup>3\*</sup>

<sup>1</sup>Department of Automation, Xiamen University, China

<sup>2</sup>Department of Computer and Information Sciences, Northumbria University, UK

<sup>3</sup>National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, China  
yushenbao@stu.xmu.edu.cn, yifeng.zeng@northumbria.ac.uk, yang@xmu.edu.cn, panyinghui@szu.edu.cn

## Abstract

Knowing a prerequisite structure among skills in a subject domain effectively enables several educational applications, including intelligent tutoring systems and curriculum planning. Traditionally, educators or domain experts use intuition to determine the skills' prerequisite relationships, which is time-consuming and prone to fall into the trap of blind spots. In this paper, we focus on inferring the prerequisite structure given access to students' performance on exercises in a subject. Nevertheless, it is challenging since students' mastery of skills can not be directly observed, but can only be estimated, i.e., its latency in nature. To tackle this problem, we propose a *causal-driven skill prerequisite structure discovery* (CSPS) method in a two-stage learning framework. In the first stage, we learn the skills' correlation relationships presented in the covariance matrix from the student performance data while, through the predicted covariance matrix in the second stage, we consider a heuristic method based on conditional independence tests and standardized partial variance to discover the prerequisite structure. We demonstrate the performance of the new approach with both simulated and real-world data. The experimental results show the effectiveness of the proposed model for identifying the skills' prerequisite structure.

## Introduction

With the burgeoning development of online learning platforms, a large amount of data on students' learning performance are accumulated, which provides a good opportunity for better education, e.g., educational data mining (EDM).

A crucial educational task in EDM is skill prerequisite structure discovery, aiming to dig out the relations among skills that have strict constraints on the order in which skills should be mastered (Chen, González-Brenes, and Tian 2016; Han, Yoon, and Yoo 2017). Obtaining the skills prerequisite structure is pivotal for a variety of educational applications, such as intelligent tutoring systems, student cognitive modeling, and curriculum planning. For example, it is superfluous to provide students with activities to solve *linear equation* exercises if they have not mastered *fractions* or other key prerequisite skills (Brunskill 2011).

Usually, building a skill order in a given subject domain is often hand-engineered by educators or experts (Liang et al.

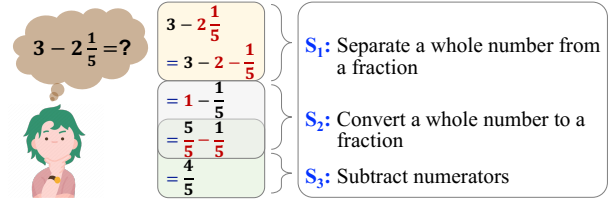


Figure 1: An example of the process to solve a *fraction-subtraction* exercise by applying three skills.

2017). But making the learning sequence choices is less than straightforward, because the experts may be susceptible to the “blind spot” (Nathan and Koedinger 2000), and some relationships may be neglected due to their non-intuitive structure (Chen, González-Brenes, and Tian 2016). Instead, some data-driven methods are made to discover prerequisite structures from student performance data, e.g., data likelihood (Brunskill 2011; Han, Yoon, and Yoo 2017), association rule mining (Chen, Willemin, and Labat 2015), and structural maximization (Chen, González-Brenes, and Tian 2016). However, such methods either focus on pairwise relationships or are labor-intensive and rarely available in practice.

Intuitively, if skill  $A$  is a prerequisite to skill  $B$ , then  $A$  can help students to learn  $B$ , thus  $A$  might be considered as a cause of  $B$ . Considering the example in Fig. 1, learning how to separate a whole number from a fraction ( $S_1$ ) can deepen the understanding of fraction properties, which further aids in mastering fraction operations, such as converting a whole number to a fraction ( $S_2$ ). Hence,  $S_1$  can be respected as one of the causes of  $S_2$ . Regardless of the interpretation, prerequisite and causal relationships share similar conditional independence in the data (Scheines, Silver, and Goldin 2014). Therefore, we can adapt techniques used for causal structure learning to discover prerequisite relationships.

Unfortunately, toward this goal, there are still some challenges. A major obstacle is that the variables of interest are latent/unmeasured. Although the relationships can be estimated by observing the student's performance on exercises that refer to these skills, most causal learning-based approaches either are originally designed to estimate causal effects between the observed variables or mainly focus on the pure measurement model (Silva et al. 2006), while in our domain, exercises usu-

\*Corresponding author.

ally load on more than one skill. In addition, data deficiency in student response logs commonly occurs because there exist some exercises that the students have never encountered.

To tackle this problem, in this paper, we propose a *causal-driven skill prerequisite structure discovery* (CSPS) model that primarily works through a two-stage framework. In the first stage, we model skills and exercises as latent variables and observed variables, which are jointly combined via a proposed *linear-Gaussian latent variable model* (LGLVM). Using Bayes' theorem, the covariance matrix of skills that implicitly contains the correlation relationships is produced. Given the covariance matrix, in the second stage, we propose a heuristic method with the causal Markov and faithfulness conditions to learn the skills' prerequisite structure. The method contains two phases, the former designs the *neighbor search* (NB-Search) algorithm, which iteratively finds out the neighbors for each skill, aiming to form the undirected skeleton of the skills' structure; and the latter proposes a *direction search based on the standardized partial variance* (DS-SPV) algorithm for orienting edges. The virtue of CSPS is that it can handle not only the data deficiency of students' response logs but also the binary scores of objective exercises and the continuous scores of subjective problems simultaneously. The main contributions are listed as follows:

- We propose a causal-driven framework (CSPS) with full structure optimization and high efficiency to infer skill prerequisite structures from students' performance data.
- We develop LGLVM to extract latent skills' partial correlation from observed performance on exercises and design the NB-Search algorithm and DS-SPV algorithm to build the skills' skeleton and orient the edges, respectively.
- We compare CSPS against all student data-based methods in recent years, including most prototypical approaches. The experimental results show the CSPS' ability to learn skill prerequisite structures.

To the best of our knowledge, this work is the first attempt to both discover the skill prerequisite structure based on causal structure learning algorithms and demonstrate the model's effectiveness in real-world data sets.

## Related Work

### Prerequisite Structure Learning

The design of data-driven methods to learn skill prerequisite structures from student performance data is explored in multiple works. Prior work by (Desmarais and Pu 2005) investigates prerequisite relationships between questions; while it does not consider the items' mapping into skills (Q-matrix). Benefiting from the Q-matrix, authors use the data likelihood (Brunskill 2011) or association rule mining (Chen, Wuillemin, and Labat 2015) to seek skills' pairwise prerequisites. Unfortunately, the local approaches fail to uncover the global structure. To address this issue, several advanced methods are proposed, including the structural maximization (Chen, González-Brenes, and Tian 2016), the Bayesian estimation (Han, Yoon, and Yoo 2017), and the restricted Bayesian inference (Saarinen, Cater, and Littman 2020). However, establishing the structure by tentatively adding

a precondition relationship between any two skills is labor-intensive and rarely available in practice.

In addition, Scheines, Silver, and Goldin (2014) introduce causal structure discovery algorithms for the prerequisite learning task. Despite the good performance in customized simulated data, it does not give an off-the-shelf solution not only because the model is not evaluated in real-world data, but also because the method can not handle cases where there are few exercises per skill. Notwithstanding, the study hints at a way for us to study the skill prerequisite relationships.

### Bayesian Network Learning and Causal Structure Discovery with Latent Variables

Historically, Bayesian networks have been useful to model skill prerequisite structures (Mislevy et al. 1999; Käser et al. 2014). Instead of learning pairwise relationships, Bayesian networks aim to model a full structure. Although discovering the structure of a Bayesian network is NP-hard (Chickering 1996), there are three types of heuristic learning techniques: constrained-based models (e.g., PC (Spirtes et al. 2000) and PC-CS (Marella and Vicard 2022)), search-and-score methods (e.g., K2 (Cooper and Herskovits 1992) and MAHC (Constantinou et al. 2022)), and hybrid approaches (e.g., MMHC (Tsamardinos, Brown, and Aliferis 2006) and CCHM (Chobtham and Constantinou 2020)).

Many learning algorithms that are designed for discrete variables can not be directly applicable in the presence of continuous variables. To remedy the void in the continuous domain, various algorithms are developed, including TC (Pellet and Elisseeff 2008), NOTEARS (Zheng et al. 2018), and DAG-NoCurl (Yu et al. 2021). The above algorithms work under the assumption that variables are Gaussian, which is a special case of the *structural equation model* (SEM) (Pearl 2000). Beyond the usual focus of structure learning among observed variables, SEM is also widely used for causal structure discovery with latent variables. In SEM, some efforts are made to discover a measurement model (Silva et al. 2006; Bandalos and Finney 2018), and others are introduced to learn relations between latent variables in a structural model (Cui et al. 2018; Rahmadi, Groot, and Heskes 2019).

### The CSPS Model

We first describe the learning task, followed by an overview of our solution. After that, we detail the CSPS model.

#### Skill Prerequisite Structure Learning Task

Given an exercise set  $\mathbf{Ex} = \{Ex_1, Ex_2, \dots, Ex_N\}$  and a skill set  $\mathbf{Sk} = \{Sk_1, Sk_2, \dots, Sk_K\}$ , we have a question matrix (Q-matrix)  $\mathbf{Q} \in \mathbb{R}^{N \times K}$ , where  $Q_{nk} = 1$  denotes  $Ex_n$  is related to  $Sk_k$ , and zero is not. Also given a student set  $\mathbf{St} = \{St_1, St_2, \dots, St_D\}$ , all students' performance on exercises are recorded in a response log, where the scores are normalized into values in  $[0, 1]$ . Note that students' performance may contain missing values since there exist some exercises that the students have never done. Given a student response log and the corresponding Q-matrix, we aim to learn potential prerequisite structures among skills.

## Framework Overview

Fig. 2 describes the process for discovering the skill prerequisite structure, and we focus on the first two stages.

**(Left)** Given the student response log and the Q-matrix (top left), we first use latent variables and observed variables to denote the skills and exercises, respectively. Next, we propose a *linear-Gaussian latent variable model* (LGLVM), in which each observation is assigned a linear Gaussian distribution based on the latent skills (middle left). The corresponding fitted parameters in the model contribute to forming the covariance matrix of skills, which sets the stage for searching the skill prerequisite structure (bottom left).

**(Middle)** Armed with the covariance matrix, for each skill, we develop a heuristic method to explore its prerequisite relations with other skills, which forms the subgraph of the prerequisite structure. The method relies on two search phases: the *skeleton search* and the *direction search*. The former finds potential neighbors for each skill and the latter determines directions between the target skill and its neighbors.

**(Right)** We construct the skill prerequisite structure by combining all the fragments, each of which is the directed subgraph of one skill with its neighbors.

## Skill Correlation Relationships Recovery

In the first stage, we formalize LGLVM, of which the parameter learning details refer to the supplemental material.

For a specific student, we cannot directly observe his/her mastered state of skills; fortunately, the student's performance on exercises provides clues to their knowledge states. Let  $\mathbf{Z} = [Z_1^\top, Z_2^\top, \dots, Z_K^\top]^\top$  be the latent variables, where  $Z_k$  is a multidimensional latent variable representing the skill  $\text{Sk}_k$ . We choose a Gaussian prior that  $Z_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{I}_k)$ , where  $\mathbf{I}_k$  is an identity matrix. We set the number of dimensions for  $Z_k$  to be  $T$ , i.e.,  $\boldsymbol{\mu}_k \in \mathbb{R}^{T \times 1}$  and  $\mathbf{I}_k \in \mathbb{R}^{T \times T}$ . Hence, the joint probability distribution of  $\mathbf{Z}$  is a multivariate Gaussian as

$$P(\mathbf{Z}) = \mathcal{N}(\mathbf{Z} | \boldsymbol{\mu}_Z, \mathbf{I}), \quad (1)$$

where  $\boldsymbol{\mu}_Z = [\boldsymbol{\mu}_1^\top, \boldsymbol{\mu}_2^\top, \dots, \boldsymbol{\mu}_K^\top]^\top$ ,  $\mathbf{I} = \text{diag}(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_K)$ .

Also let  $\mathbf{O} = [O_1, O_2, \dots, O_N]^\top$  be the observed variables, where  $O_n$  is the observed variable of a student's performance on  $\text{Ex}_n$ . Based on the latent variables  $\mathbf{Z}$ , the joint probability distribution of the observed variables  $\mathbf{O}$  is built by assuming that there exists a linear mapping from the space describing the skills to the numerical performance scale, which gives

$$P(\mathbf{O} | \mathbf{Z}) = \mathcal{N}(\mathbf{O} | \mathbf{Q}\mathbf{O}\mathbf{W}\mathbf{Z} + \boldsymbol{\Phi}, \boldsymbol{\Theta}). \quad (2)$$

In Eq. (2),  $\mathbf{W}$  is a loading matrix, where  $\mathbf{W}_{nk}$  is the  $1 \times T$  vector, indicating the average proficiency level of students on  $\text{Sk}_k$  with respect to  $\text{Ex}_n$ . Note that  $\mathbf{W}_{nk} \neq \mathbf{0}$  if and only if  $\mathbf{Q}_{nk} \neq 0$ .  $\boldsymbol{\Phi} = [\phi_1, \phi_2, \dots, \phi_N]^\top$  is a constant vector, where  $\phi_n$  represents the average complexity of  $\text{Ex}_n$ .  $\boldsymbol{\Theta} \in \mathbb{R}^{N \times N}$  is a diagonal matrix, where the  $n$ -th element on the diagonal is  $\theta_n$ . Given an observed variable  $O_n$ , we have

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

Again, we need to underline that given  $P(\mathbf{Z})$ ,  $P(\mathbf{O}|\mathbf{Z})$  has a mean that is a linear function of  $\mathbf{Z}$ , and a covariance being

independent of  $\mathbf{Z}$ . Armed with  $P(\mathbf{Z})$  and  $P(\mathbf{O}|\mathbf{Z})$ , using the Bayes' theorem, the conditional distribution  $P(\mathbf{Z}|\mathbf{O})$  is also Gaussian, which can be expressed as

$$P(\mathbf{Z}|\mathbf{O}) = \mathcal{N}(\mathbf{Z} | \boldsymbol{\mu}_Z^c, \boldsymbol{\Sigma}^c),$$

where  $\boldsymbol{\mu}_Z^c$  and  $\boldsymbol{\Sigma}^c$  are the conditional mean vector and covariance matrix, respectively. Here,  $\boldsymbol{\Sigma}^c$  is given by

$$\boldsymbol{\Sigma}^c = [\mathbf{I}^{-1} + (\mathbf{Q}\mathbf{O}\mathbf{W})^\top \boldsymbol{\Theta}^{-1} (\mathbf{Q}\mathbf{O}\mathbf{W})]^{-1}. \quad (3)$$

It is worth taking a moment to study the covariance matrix in Eq. (3). Before observing the student performance data (i.e.,  $\mathbf{O}$ ), the skill relations remain unknown, so we assume that the latent variables are mutually independent, i.e., the identity covariance matrix in Eq. (1). After the observation  $\mathbf{O} = \mathbf{o}$ , we obtain an updated distribution for  $\mathbf{Z}$ , where Eq. (3) determines the updated covariance matrix  $\boldsymbol{\Sigma}^c$  of the updated distribution. Note that  $\boldsymbol{\Sigma}^c$  is non-diagonal, which indicates the observations  $\mathbf{O}$  offer the information of the correlation relationships among skills via  $\mathbf{W}$  and  $\boldsymbol{\Theta}$ .

## Skill Prerequisite Structure Learning

In the second stage, we turn the spotlight on modeling the skill prerequisite structure. To this end, we design a heuristic method in a two-phase process under the causal Markov condition and the faithfulness assumption, i.e., the skeleton search (**Phase I**) and the direction search (**Phase II**).

**Phase I: Skeleton Search.** The skeleton search aims to identify the undirected subgraph for latent skills, which is based on the proposed local discovery algorithm called NB-search. The NB-search on target latent variable  $Z_k$  provides a way to select potential neighbors (the set of parents and children) for  $Z_k$ , denoted  $\text{NB}(Z_k)$ . By employing the NB-search with every latent variable as the target one, we can identify all the edges (in an unoriented fashion) for the skills' network, i.e., identifying a skeleton of the skill prerequisite structure.

The basic idea of the NB-search is to invoke a function for testing whether the target variable  $Z_k$  is (conditionally) independent of  $Z_i$  ( $Z_i \neq Z_k$ ) given the subset  $\mathcal{Y} \subseteq \mathbf{Z}$ , i.e.,  $Z_i \perp\!\!\!\perp Z_k | \mathcal{Y}$  holds. The function is given by (Tsamardinos, Brown, and Aliferis 2006)

$$\text{Assoc}(Z_i, Z_k | \mathcal{Y}) = 1 - p(Z_i, Z_k | \mathcal{Y}),$$

where  $\text{Assoc}(Z_i, Z_k | \mathcal{Y})$  is an estimation of the strength of association (dependency) of  $Z_i$  and  $Z_k$  given  $\mathcal{Y}$ , and we assume that  $Z_i \perp\!\!\!\perp Z_k | \mathcal{Y} \Leftrightarrow (\text{Assoc}(Z_i, Z_k | \mathcal{Y}) = 0)$ .  $p(Z_i, Z_k | \mathcal{Y})$  is the  $p$ -value returned by the *Fisher's z*-test, which uses the covariance matrix  $\boldsymbol{\Sigma}^c$  in Eq. (3)<sup>1</sup>. It also can be seen that  $\text{Assoc}(\cdot)$  ranges from 0 to 1, and the value increases monotonically with the decreasing of the  $p$ -value, and reaches a peak as the  $p$ -value is zero, which indicates the strongest connection between the variables. Given the target latent variable of interest  $Z_k$  ( $1 \leq k \leq K$ ), we present the pseudocodes of the NB-search procedure in Algorithm 1 (Tsamardinos, Brown, and Aliferis 2006).

<sup>1</sup>We treat the estimated latent covariance matrix (i.e.,  $\boldsymbol{\Sigma}^c$  in Eq. (3)) as a sample correlation matrix among observed variables, and then use  $\boldsymbol{\Sigma}^c$  to make decisions about whether the (conditional) independence constraints over  $\boldsymbol{\Sigma}^c$  hold. For this, we intuitively assume that the sample size of the estimated covariance matrix is  $D$ , which lacks statistical guarantees notwithstanding.

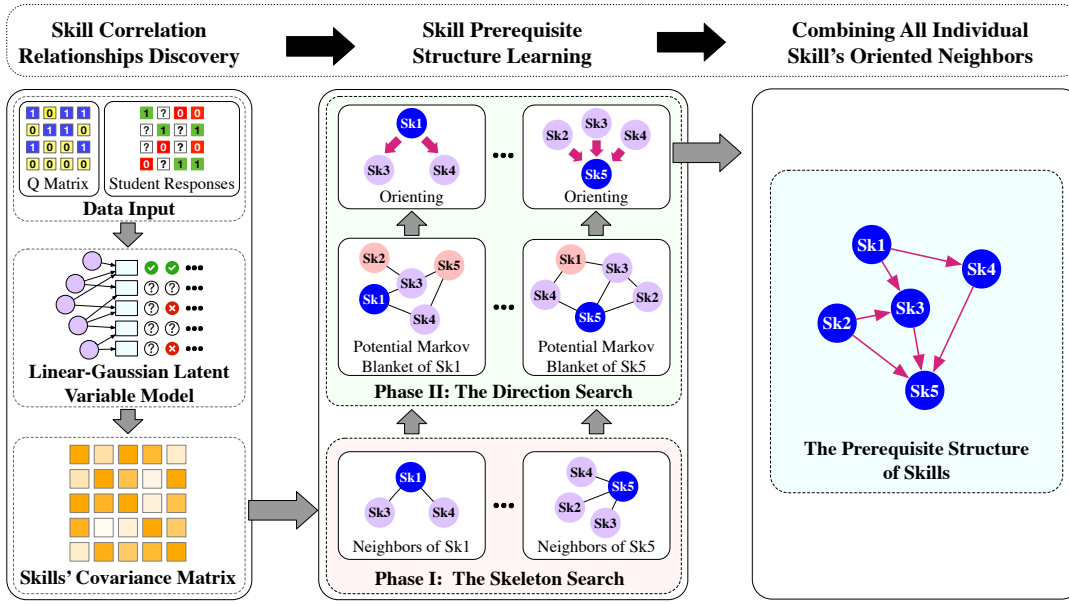


Figure 2: The new framework includes (a) a linear-Gaussian latent variable model, (b) the skill prerequisite structure search via two phases, and (c) a combination of the learned subgraphs.

---

Algorithm 1: The NB-Search Algorithm

---

**Input:**  $Z_k, \mathbf{Z}, \Sigma^c$   
**Output:**  $\mathbf{NB}(Z_k)$

- 1: Initialize  $\mathbf{CS} = \emptyset, \mathbf{NB} = \emptyset$
- 2: **for**  $i \leftarrow 1$  to  $K$ , and  $i \neq k$  **do**
- 3:   Calculate  $\mathbf{Assoc}(Z_i, Z_k)$
- 4:    $\mathbf{CS} \leftarrow \mathbf{CS} \cup \{Z_i\}$
- 5: **end for**
- 6: **while**  $\mathbf{CS} \neq \emptyset$  **do**
- 7:   Find  $Z_i \in \mathbf{CS}$  that maximizes the  $\mathbf{Assoc}(Z_i, Z_k)$  value
- 8:    $\mathbf{NB} \leftarrow \mathbf{NB} \cup \{Z_i\}$
- 9:   **if**  $\exists Z_j \in \mathbf{NB}$  and  $\mathcal{Y} \subseteq \mathbf{NB} \setminus \{Z_j, Z_k\}$ , s.t.  $Z_j \perp\!\!\!\perp Z_k | \mathcal{Y}$  **then**
- 10:      $\mathbf{NB} \leftarrow \mathbf{NB} \setminus \{Z_j\}$  and do not consider  $Z_j$
- 11:   **end if**
- 12:    $\mathbf{CS} \leftarrow \mathbf{CS} \setminus \{Z_i\}$
- 13: **end while**
- 14: **return**  $\mathbf{NB}(Z_k)$

---

**Phase II: Direction Search.** After identifying the potential neighbors of  $Z_k$  ( $1 \leq k \leq K$ ), a second critical step is determining the directions between  $Z_k$  and its neighbors  $\mathbf{NB}(Z_k)$ . Here, we use a heuristic to orient the edges by multiple tests of the log-ratio of the SPV (Opgen-Rhein and Strimmer 2007), which measures the proportion of the variance that remains by regressing against all other related variables.

Before calculating the SPV value of  $Z_k$ , we need to find all variables that may influence  $Z_k$ . In this paper, we consider the potential Markov blanket of  $Z_k$ , denoted  $\mathbf{PMB}(Z_k)$ . The important bonus of the Markov blanket is that it stores all the information that can not be obtained from any other variables (Pellet and Elisseeff 2008), which covers all inde-

pendent variables of  $Z_k$  that are to be regressed away. Hence, the SPV value of  $Z_k$  can be calculated as follows

$$\text{SPV}_{Z_k} = \frac{\sigma_{Z_k \cdot \mathbf{PMB}(Z_k)}^2}{\sigma_{Z_k}^2}, \quad (4)$$

where  $\sigma_{Z_k}^2$  is the variance of  $Z_k$ ,  $\sigma_{Z_k \cdot \mathbf{PMB}(Z_k)}^2$  is the partial variance of  $Z_k$  given all variables in  $\mathbf{PMB}(Z_k)$ . Again, the two values can be obtained via  $\Sigma^c$  in Eq. (3).

Based on the SPV, we propose the DS-SPV algorithm to search the directions between the target  $Z_k$  and its neighbors  $\mathbf{NB}(Z_k)$ , which contributes to form the (partial) directed acyclic graph of  $Z_k$ , denoted  $\mathcal{G}(Z_k)$  (see Algorithm 2). As observed, the DS-SPV begins with an empty graph  $\mathcal{G}(Z_k)$ , and then adds nodes including  $Z_k$  and its neighbors into  $\mathcal{G}(Z_k)$ <sup>2</sup> (lines 1-2). Next, the DS-SPV calls Algorithm 3 to detect a potential Markov blanket of  $Z_k$  (line 3), which is followed by calculating the SPV value of  $Z_k$  using Eq. (4) (line 4). After this, the algorithm repeatedly compares the SPV value between  $Z_k$  and its neighbors (lines 5-16). Specifically, we use the log-ratio  $\log(\text{SPV}_{Z_k}/\text{SPV}_{Z_i})$  to determine the edge direction between  $Z_i$  and  $Z_k$ . All edges are directed in such a fashion that the direction of the arrow from  $Z_k$  to  $Z_i$  if  $\log(\text{SPV}_{Z_k}/\text{SPV}_{Z_i}) > 0$  (lines 9-10), which means that we impose the directionality from the more “exogenous” variable to the more “endogenous” variable. Similarly, if  $\log(\text{SPV}_{Z_k}/\text{SPV}_{Z_i}) < 0$ , the direction is from  $Z_i$  to  $Z_k$  (lines 11-12). Note that the other edges with  $\log(\text{SPV}_{Z_k}/\text{SPV}_{Z_i}) \approx 0$  remain undirected (lines 13-14), it can be interpreted as the existence of the mutual influence of the two variables in the underlying regression system. The

<sup>2</sup>In the context of the skills’ prerequisite structure, if not otherwise specified, variables are also interchangeably called nodes.

**Algorithm 2: The DS-SPV Algorithm****Input:**  $Z_k, \mathbf{NB}(Z_k), \Sigma^c$ **Output:**  $\mathcal{G}(Z_k)$ 


---

```

1: Initialize  $\mathcal{G}(Z_k) = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \emptyset, \mathcal{E} = \emptyset$ 
2:  $\mathcal{V} \leftarrow \{Z_k\} \cup \mathbf{NB}(Z_k)$ 
3:  $\mathbf{PMB}(Z_k) \leftarrow \text{PMB-Search}(Z_k, \mathbf{NB}(Z_k))$ 
4: Calculate  $\text{SPV}_{Z_k}$  by Eq. (4)
5: for every neighbor  $Z_i \in \mathbf{NB}(Z_k)$  do
6:    $\mathbf{NB}(Z_i) \leftarrow \text{NB-Search}(Z_i)$ 
7:    $\mathbf{PMB}(Z_i) \leftarrow \text{PMB-Search}(Z_i, \mathbf{NB}(Z_i))$ 
8:   Calculate  $\text{SPV}_{Z_i}$  by Eq. (4)
9:   if  $\log(\text{SPV}_{Z_k}/\text{SPV}_{Z_i}) > 0$  then
10:    Add the directed edge  $Z_k \rightarrow Z_i$  to  $\mathcal{E}$ 
11:   else if  $\log(\text{SPV}_{Z_k}/\text{SPV}_{Z_i}) < 0$  then
12:    Add the directed edge  $Z_i \rightarrow Z_k$  to  $\mathcal{E}$ 
13:   else
14:    Add the undirected edge  $Z_i - Z_k$  to  $\mathcal{E}$ 
15:   end if
16: end for
17: return  $\mathcal{G}(Z_k)$ 

```

---

**Algorithm 3: The PMB-Search Algorithm****Input:**  $Z_k, \mathbf{NB}(Z_k)$ **Output:**  $\mathbf{PMB}(Z_k)$ 


---

```

1: Initialize  $\mathbf{PMB}(Z_k) = \emptyset$ 
2: for every neighbor  $Z_i \in \mathbf{NB}(Z_k)$  do
3:    $\mathbf{NB}(Z_i) \leftarrow \text{NB-Search}(Z_i)$ 
4:    $\mathbf{PMB}(Z_k) \leftarrow \mathbf{PMB}(Z_k) \cup \mathbf{NB}(Z_i)$ 
5: end for
6: return  $\mathbf{PMB}(Z_k)$ 

```

---

iterations are repeated until we finish testing all the log ratios, and the partially directed acyclic graph of  $Z_k$  with its neighbors is returned (line 17).

**Complexity Analysis**

We discuss the time complexity of CSPS, which involves two stages. In the first stage, the total time complexity is  $O(\#iter \times (\text{DNK}^2))$ , and the second stage is bounded by  $O(K^2 \times (|\mathbf{NB}|^4 + K^2))$ , where  $\#iter$  is the number of iterations for convergence, and  $\mathbf{NB}$  is the largest set of neighbors over all variables. The proposed model is efficient since we construct the full structure by only considering the Markov blanket of variables and using local learning operations that capture (conditional) independence between skills. The analysis details refer to the supplemental material.

**Experimental Study**

We evaluate the effectiveness of CSPS in recovering the prerequisite structure of skills with both simulated data (Sync1 and Sync2) and real-world data (FrcSub and Alg0506).

**Experimental Setup**

**Evaluation Metrics.** Evaluating the prerequisite structure is non-trivial (Vuong, Nixon, and Towle 2011), here, we use

the metrics that are designed for Bayesian network structure learning (Xie et al. 2020). Specifically, we use the *adjacency score* ( $F_1$ -AR) and *orientation score* ( $F_1$ -OR), which measure how well we can recover the connections between nodes and directions of the edges, respectively (Chen, González-Brenes, and Tian 2016). In both cases, the score reaches its best value at 1 and worst at 0. The formulas are

$$\text{TPAR} = \frac{\# \text{of correct adjacencies in learned network}}{\# \text{of adjacencies in true network}},$$

$$\text{TDAR} = \frac{\# \text{of correct adjacencies in learned network}}{\# \text{of adjacencies in learned network}},$$

$$F_1\text{-AR} = 2 \times \frac{\text{TPAR} \times \text{TDAR}}{\text{TPAR} + \text{TDAR}},$$

$$\text{TPOR} = \frac{\# \text{of correctly directed edges in learned network}}{\# \text{of directed edges in true network}},$$

$$\text{TDOR} = \frac{\# \text{of correctly directed edges in learned network}}{\# \text{of directed edges in learned network}},$$

$$F_1\text{-OR} = 2 \times \frac{\text{TPOR} \times \text{TDOR}}{\text{TPOR} + \text{TDOR}}.$$

**Baseline Approaches.** We consider baseline approaches as:

- **EPS-ND** (Brunskill 2011) aims to discover the plausible precondition relationship among each pair of skills.
- **CITS** (Scheines, Silver, and Goldin 2014) models the prerequisite relations among skills as a causal graph via the PC algorithm (Spirtes et al. 2000).
- **PARM** (Chen, Willemin, and Labat 2015) is a data mining approach to discover the (pair) relations of skills.
- **CMPD** (Chen, González-Brenes, and Tian 2016) learns the skill prerequisite relations using the Bayesian network structure learning technique.
- **BE-NMC** (Han, Yoon, and Yoo 2017) uses the likelihood ratio test based on the pseudo-Bayes factor to explore the prerequisite relationship between given two skills.
- **DIDACT** (Saarinen, Cater, and Littman 2020) is a fast discrete model that blends the expressive capability and dependency modeling from the Bayesian network with the fast inference of item response theory.

**Implementation Details.** In the following experiments, all the numerical computations are conducted on an Ubuntu server with a Core i9-1090K 3.7 GHz and 128 GB memory. In terms of the baseline implementations, we use the following protocol: (a) The authors' implementation is preferred; (b) If the condition is not met, we employ the best publicly available implementation of the model. (c) Otherwise, we re-implement the model and use our version. For example, we use the *semopy* (Igolkina and Meshcheryakov 2020) Python package to estimate all structural equation models required in the CITS, and the PC algorithm is implemented in the Python package *pcalg*. Unless stated, we report the average experimental results over 10 repeated trials.

**Evaluation**

**Simulated Test Data.** We first conduct a simulated study to evaluate the effectiveness of CSPS. For this, we engineer



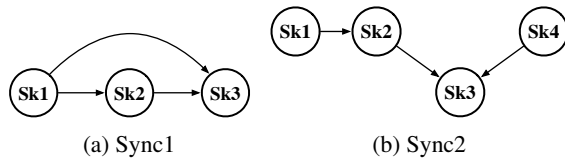


Figure 3: Two skill structures for generating the simulation data. Exercise nodes are omitted for conciseness.

two prerequisite structures, each of which encodes different causal relations between skills in Fig. 3. Since all baselines except for the CITS are restricted to the implementation with discrete variables, for each structure, we consider the simulated binary data with different numbers of observations (i.e.,  $D = 150, 500, 1000, 2000$ )<sup>3</sup>. Thus, we generate 8 synthetic data sets, i.e., 2 structures  $\times$  4 sample sizes.

Table 1 and Table 2 provide the final results, where “-” denotes no edges in the learned graph. Here we use the boldface to highlight the best results, and the top 2 performances are in the shade. As observed, CSPS consistently obtains the top-2 performance in terms of the adjacency score ( $F_1$ -AR) and orientation score ( $F_1$ -OR) in most sample-size cases, especially achieving the highest value of  $F_1$ -AR when the sample sizes are 500 and 2000. Further observations reveal that Bayesian inference methods including CMPD, BE-NMC, and DIDACT perform well in both data sets. As for PARM, the rule-based model performs poorly, we guess the possible reason is that PARM is limited to discovering pair-wise prerequisite relationships, instead of constructing the full structure. Taken together, these results indicate the effectiveness of CSPS in discovering the skills’ prerequisite structure.

**Real-World Test Data.** We turn to evaluate CSPS using the real-world test data (FrcSub), which is made of binary test responses (right or wrong) of 535 examinees on 20 *fraction-subtraction* exercises measuring 8 skills. For a better illustration, Fig. 4 provides the prerequisite structure of the skills.

We compare CSPS with the baseline approaches in Fig. 5. Considering the adjacency score first (Fig. 5a), CSPS shows the best performance in discovering the adjacencies (see  $F_1$ -AR). The value of TPAP achieves 0.84, which means that we recover 84% adjacencies for FrcSub. Note that EPS-ND and PARM perform well in terms of TPAP, however, they pay a price for the highest true positive adjacency rate because their values of TDAR are lower, which means that EPS-ND and PARM tend to allow for a lot of redundant edges, blurring the true prerequisite structure. Besides, it is apparent from Fig. 5a that CITS fails to find out the true prerequisite relationships as the TPAP approaches zero. The poor results also confirm the limitation mentioned in (Chen, González-Brenes, and Tian 2016), that is, the technique is not suitable for the real-world data featured in complexity and variability.

Similar results also hold in terms of the orientation score in Fig. 5b, which shows our model possesses a satisfying ability for edge orientations. However, all models do not perform well compared with the performance in adjacency

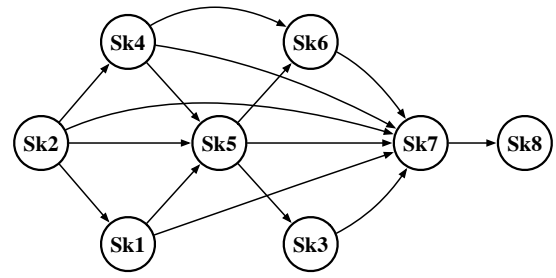


Figure 4: Skill prerequisite structure of the FrcSub.

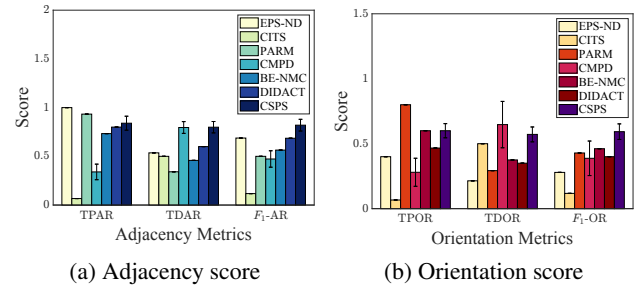


Figure 5: Comparison of CSPS with the baseline approaches for discovering the prerequisite relationships in FrcSub.

score. There are two main reasons: first, an edge can only be correctly oriented if the adjacency has been successfully discovered; second, since the orientation decisions involve interaction between nearby edges, the adjacency errors from the nearby edges also produce additional orientation errors. Thus, in this sense, orientation is more difficult than adjacency.

**Real-World Log Data.** We now evaluate CSPS with the public real-world log data of the 2005-2006 curriculum “Algebra I”<sup>4</sup>, abbreviated as Alg0506, which is collected from the interactions between students and computer-aided tutoring systems. After the data processing, the pre-processed data set includes 438 students, 185 exercises, and 12 skills, and the prerequisite structure of skills is shown in Fig. 6. Note that the data density of the student response log is 21.05%, which means that there are approximately 80% of the exercises that each student has never done.

In the pre-processed Alg0506, since all baselines except for CITS are not applicable due to the observations being continuous, we only experiment on the structure discovery performance between CSPS and CITS, and Fig. 7 shows the comparison results. From Fig. 7, we have the following observations:

- (a) The CSPS model obtains good performance for the discovery of the prerequisite structure (see the adjacency score in Fig. 7a and the orientation score in Fig. 7b, respectively);
- (b) Compared with the performance of CITS in FrcSub, the causal-based model yields similar results, which further confirm that the CITS is not applicable to real-world data; and (c) In FrcSub, CSPS yields nearly 80% and 60% of  $F_1$ -AR and  $F_1$ -OR, of which are only 70% and 45% in Alg0506. The results can be explained by:

<sup>3</sup>We also compare CSPS with CITS in the continuous version of the data sets (see supplemental materials).

<sup>4</sup><https://pslcdatashop.web.cmu.edu/KDDCup/>

Model	Sample Size (Sync1)				Sample Size (Sync2)			
	150	500	1000	2000	150	500	1000	2000
EPS-ND	<b>1.000±.000</b>	<b>1.000±.000</b>	<b>1.000±.000</b>	<b>1.000±.000</b>	0.667±.000	0.667±.000	0.667±.000	0.667±.000
CITS	-	0.800±.000	0.667±.000	0.667±.000	0.120±.193	0.571±.000	<b>0.857±.000</b>	0.667±.000
PARM	0.667±.000	0.667±.000	0.667±.000	0.667±.000	0.461±.000	0.444±.000	0.462±.000	0.667±.000
CMPD	0.800±.183	0.980±.063	0.980±.063	<b>1.000±.000</b>	0.753±.078	0.732±.056	0.699±.061	0.725±.040
BE-NMC	0.667±.000	0.750±.000	0.667±.000	0.667±.000	<b>1.000±.000</b>	0.667±.000	<b>1.000±.000</b>	0.462±.000
DIDACT	<b>1.000±.000</b>	0.667±.000	0.667±.000	0.667±.000	0.600±.000	<b>0.750±.000</b>	0.600±.000	0.667±.000
CSPS	0.930±.163	<b>1.000±.000</b>	<b>1.000±.000</b>	<b>1.000±.000</b>	0.500±.163	<b>0.750±.000</b>	0.750±.000	<b>0.750±.000</b>

Table 1: Comparison of CSPS with the baselines in terms of  $F_1$ -AR

Model	Sample Size (Sync1)				Sample Size (Sync2)			
	150	500	1000	2000	150	500	1000	2000
EPS-ND	0.000±.000	0.667±.000	0.667±.000	<b>1.000±.000</b>	0.444±.000	0.444±.000	0.444±.000	0.444±.000
CITS	-	0.000±.000	0.667±.000	0.667±.000	0.120±.193	0.286±.000	0.286±.000	0.444±.000
PARM	0.667±.000	0.667±.000	0.667±.000	0.667±.000	0.461±.000	0.444±.000	0.462±.000	<b>0.667±.000</b>
CMPD	0.577±.310	<b>0.740±.252</b>	0.813±.228	0.640±.237	0.182±.126	0.404±.125	0.431±.091	0.483±.027
BE-NMC	0.800±.000	0.500±.000	0.667±.000	0.667±.000	0.333±.000	<b>0.667±.000</b>	0.333±.000	0.462±.000
DIDACT	<b>1.000±.000</b>	0.667±.000	0.667±.000	0.667±.000	<b>0.600±.000</b>	0.500±.000	0.600±.000	<b>0.667±.000</b>
CSPS	0.623±.094	0.667±.000	<b>0.900±.061</b>	0.667±.000	0.450±.227	0.500±.000	<b>0.700±.105</b>	0.500±.000

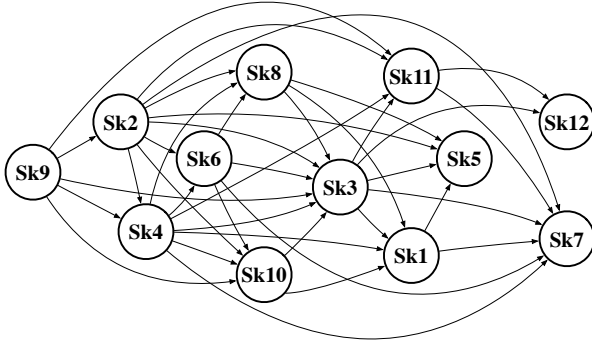
Table 2: Comparison of CSPS with the baselines in terms of  $F_1$ -OR

Figure 6: Skill prerequisite structure of the Alg0506.

- Alg0506 contains a lot of noise. We observe that some students may wander around several steps by repeatedly using some skills. Although the students finished these exercises finally, the hesitancy impairs the inference of the true prerequisite relationships.
- There exist some exercises, where the skills are error-tagging or miss-labeled.
- For complex exercises, there may be multiple solutions.
- Some skills could be mastered if the students have taken sufficient training, even though some prerequisites are not previously learned (Chen, Wullemmin, and Labat 2015).

## Conclusion

In this paper, we provide a focused study on discovering the skill prerequisite structure from student performance data.

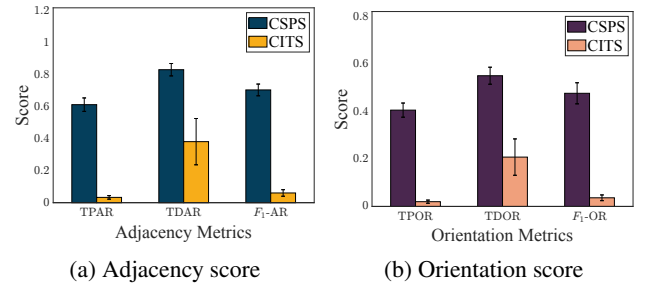


Figure 7: Performance comparison of CSPS with CITS in Alg0506. We do not give the results of the other models because the observations in Alg0506 are continuous.

Here, we take inspiration from techniques originally designed for learning causal structures. However, the variables of interest are latent in nature, and it is not immune to the uncertainty in measuring the mastery status of students' skills from performance data. To tackle this problem, we introduce a model based on the causal structure learning technique namely CSPS. CSPS works through a two-stage framework, i.e., skill correlation relationships recovery and skill prerequisite structure learning. We evaluate the CSPS on both the simulated and real-world data. Experimental results demonstrate the ability of our model to recover skills prerequisite structures. Further research can be conducted on investigating efficient search methods in the context of having more latent variables, as long as we collect enough exercise data of students in the knowledge-learning settings.

## Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grants No.62176225 and 62276168) and the Natural Science Foundation of Guangdong Province, China(Grant No. 2023A1515010869).

## References

- Bandalos, D. L.; and Finney, S. J. 2018. Factor analysis: exploratory and confirmatory. In *the Reviewer's Guide to Quantitative Methods in the Social Sciences*, 98–122. Routledge.
- Brunskill, E. 2011. Estimating prerequisite structure from noisy data. In *Proceedings of the 4th International Conference on Educational Data Mining*, 217–222.
- Chen, Y.; González-Brenes, J. P.; and Tian, J. 2016. Joint discovery of skill prerequisite graphs and student models. In *Proceedings of the 9th International Conference on Educational Data Mining*, 46–53.
- Chen, Y.; Wuillemin, P.-H.; and Labat, J.-M. 2015. Discovering prerequisite structure of skills through probabilistic association rules mining. In *Proceedings of the 8th International Conference on Educational Data Mining*, 117–124.
- Chickering, D. M. 1996. Learning Bayesian networks is NP-complete. In *Learning from Data – 5th International Workshop on Artificial Intelligence and Statistics*, 121–130. Springer.
- Chobtham, K.; and Constantinou, A. C. 2020. Bayesian network structure learning with causal effects in the presence of latent variables. In *International Conference on Probabilistic Graphical Models*, 101–112. PMLR.
- Constantinou, A. C.; Liu, Y.; Kitson, N. K.; Chobtham, K.; and Guo, Z. 2022. Effective and efficient structure learning with pruning and model averaging strategies. *International Journal of Approximate Reasoning*, 151: 292–321.
- Cooper, G. F.; and Herskovits, E. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4): 309–347.
- Cui, R.; Groot, P.; Schauer, M.; and Heskes, T. 2018. Learning the causal structure of copula models with latent variables. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, 188–197.
- Desmarais, M. C.; and Pu, X. 2005. A Bayesian student model without hidden nodes and its comparison with item response theory. *International Journal of Artificial Intelligence in Education*, 15(4): 291–323.
- Han, S.-Y.; Yoon, J.; and Yoo, Y. J. 2017. Discovering skill prerequisite structure through Bayesian-estimation and nested model comparison. In *Proceedings of the 10th International Conference on Educational Data Mining*, 398–399.
- Igolkina, A. A.; and Meshcheryakov, G. 2020. Semopy: a Python package for structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, 0(0): 1–12.
- Käser, T.; Klingler, S.; Schwing, A. G.; and Gross, M. 2014. Beyond knowledge tracing: modeling skill topologies with bayesian networks. In *International Conference on Intelligent Tutoring Systems*, 188–198.
- Liang, C.; Ye, J.; Wu, Z.; Pursel, B.; and Giles, C. L. 2017. Recovering concept prerequisite relations from university course dependencies. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 4786–4791.
- Marella, D.; and Vicard, P. 2022. Bayesian network structural learning from complex survey data: a resampling based approach. *Statistical Methods & Applications*, 1–33.
- Mislevy, R. J.; Almond, R. G.; Yan, D.; and Steinberg, L. S. 1999. Bayes nets in educational assessment: where the numbers come from. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 437–446.
- Nathan, M. J.; and Koedinger, K. R. 2000. An investigation of teachers' beliefs of students' algebra development. *Cognition and Instruction*, 18(2): 209–237.
- Opgen-Rhein, R.; and Strimmer, K. 2007. From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC Systems Biology*, 1(1): 1–10.
- Pearl, J. 2000. Models, reasoning and inference. *Cambridge University Press*, 19: 2.
- Pellet, J.-P.; and Elisseeff, A. 2008. Using Markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9(7).
- Rahmadi, R.; Groot, P.; and Heskes, T. 2019. Stable specification search in structural equation models with latent variables. *ACM Transactions on Intelligent Systems and Technology*, 10(5): 1–23.
- Saarinen, S.; Cater, E.; and Littman, M. L. 2020. Applying prerequisite structure inference to adaptive testing. In *Proceedings of the 10th International Conference on Learning Analytics & Knowledge*, 422–427.
- Scheines, R.; Silver, E.; and Goldin, I. M. 2014. Discovering prerequisite relationships among knowledge components. In *Proceedings of the 7th International Conference on Educational Data Mining*, 355–356.
- Silva, R.; Scheines, R.; Glymour, C.; Spirtes, P.; and Chickering, D. M. 2006. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7(2).
- Spirtes, P.; Glymour, C. N.; Scheines, R.; and Heckerman, D. 2000. *Causation, prediction, and search*. MIT press.
- Tsamardinos, I.; Brown, L. E.; and Aliferis, C. F. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1): 31–78.
- Vuong, A.; Nixon, T.; and Towle, B. 2011. A method for finding prerequisites within a curriculum. In *Proceedings of the 4th International Conference on Educational Data Mining*, 211–216.
- Xie, F.; Cai, R.; Zeng, Y.; Gao, J.; and Hao, Z. 2020. An efficient entropy-based causal discovery method for linear structural equation models with iid noise variables. *IEEE Transactions on Neural Networks and Learning Systems*, 31(5): 1667–1680.
- Yu, Y.; Gao, T.; Yin, N.; and Ji, Q. 2021. DAGs with no curl: an efficient DAG structure learning approach. In *International Conference on Machine Learning*, 12156–12166. PMLR.



Zheng, X.; Aragam, B.; Ravikumar, P. K.; and Xing, E. P. 2018. Dags with no tears: continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31.