Discovering Prerequisite Relationships among Knowledge Components

Conference Paper · January 2014

3 authors, including:

Ilya Goldin
Phenom People
31 PUBLICATIONS   **280** CITATIONS

Some of the authors of this publication are also working on these related projects:

Individual Differences View project

Domain Modeling View project

Discovering Prerequisite Relationships among Knowledge Components

Richard Scheines
Department of Philosophy
Carnegie Mellon University
scheines@cmu.edu

Elizabeth Silver
Department of Philosophy
Carnegie Mellon University
silver@cmu.edu

Ilya Goldin
Center for Digital Data, Analytics, and
Adaptive Learning
Pearson Education
ilya.goldin@pearson.com

ABSTRACT

Knowing the prerequisite structure among the knowledge components in a domain is crucial for designing instruction and for assessing mastery. Treating KCs as latent variables, we investigate how data on the items that test these skills can be used to discover the prerequisite structure among such skills. Our method assumes that we know or have discovered the Q-matrix (the measurement model) that connects latents representing the skill to items measuring the skills. By modeling the pre-requisite relations as a causal graph, we can then search for the causal structure among the latents via an extension of an algorithm introduced by Spirtes, Glymour, and Scheines in 2000. We validate the algorithm using simulated data, and discuss a potential application to a High School geometry assessment dataset.

Keywords

Domain models, knowledge components, q-matrix, prerequisites, causal discovery

1. INTRODUCTION

Instructors, human or automated, must often choose the order in which to present topics to a student, and self-taught learners must choose for themselves the order in which to study the topics of a new domain. However, making these ordering choices can be less than straightforward. For example, should one study the computation of the area of a square before or after computation of the area of a rectangle? Perhaps the square is a simple way of introducing the notion of area, and this introduction facilitates subsequent generalization to learning the rectangle formula. Or perhaps the rectangle is a useful general notion, from which the square falls out as a special case. Thus, it may be effective to study either topic first, or it may be that a specific topic ordering is preferable. Specifically, we need to determine the prerequisite structure of a domain. [4,5]

How can we choose a topic ordering? It seems obvious to ask an expert. But just as an instructor or researcher may hold an "expert blind spot" regarding which topic is more difficult for learners [6], we suspect that expert opinions are not a reliable way to determine effective topic order. Besides, there are a great many topics in each domain, and asking experts is prohibitively costly in time an effort. Instead, we turn to discovering prerequisite structures from data.

Broadly, the discovery of prerequisite structures has two aspects: discovering the topics and ordering the topics. The former is an active area of research. Modern methods involve the discovery of the Q-matrix, which specifies the mapping between the tasks that students perform (i.e. items, in the language of psychometrics) and the skills that the tasks require. [7,8,9] But having discovered the Q-matrix, in what order should we present its component skills to the learner? This issue, ordering the topics, has been explored only sparsely in the literature.

A recent contribution on topic ordering is due to researchers at Carnegie Learning [10], who applied a test to data on almost every possible instructional unit pairing in four Carnegie Learning math curricula. The test relied on natural variation in longitudinal data collected from many instructors' use of the Cognitive Tutor to see if students could succeed on a unit without having earlier mastered another unit.

Notably, the unit is not the smallest level of organization where prerequisite structure matters. "Units cover distinct mathematical topics; sections [within units] cover distinct sets of problems on that topic, with a distinct student skill model for each section." [10] The ordering of sections may also have instructional implications, which we consider in this work.

By contrast, our work is based not on longitudinal data, but data from a single assessment administered to multiple students at a single point in time. As a motivating case, we considered a test administered to approximately 120 students in a developmental mathematics course (a course meant to address gaps in math preparedness for students enrolling in college). Our research question was: Could we infer prerequisite structure based on variation in student performance on an assessment?

Our method of prerequisite discovery applies to an assessment of any scope, regardless of whether it covers multiple problem-solving strategies on a skill, or multiple skills on a single learning objective, or multiple objectives in a syllabus, or multiple courses in a multi-year curricular sequence (e.g., a standardized test).

Our approach is based on causal structure discovery algorithms. Intuitively, if a student knows a prerequisite A for skill B, then A can help her to learn B, so A might be considered a cause of B. Regardless of the interpretation, prerequisite and causal relationships should produce similar conditional independences in the data. Prerequisite relationships between skills should produce correlations between related skills, as students who have mastered the prerequisites for a given skill are more likely to have mastered the skill itself. Furthermore, prerequisite relationships should produce "screening off" effects. For example: if A is a prerequisite for B, and B for C, but A is not *directly* required for C, then if I know a student has mastered B, learning that she has also mastered A will not inform me about whether she has mastered C. We can therefore adapt techniques originally designed for learning causal structures to discover prerequisite relationships, using data collected at a single point in time.

However, since the skills are not directly observable, learning the relationships between them is difficult. There is already a method (a combination of two algorithms, Build Pure Clusters (BPC)[1][17] and MIMbuild [3, page 319]) for discovering causal structure in the case where the Q-matrix is unknown, but contains many *pure items* (i.e. items that load on only one skill, or latent variable). Unfortunately, in our target applications, most test items load on at least two distinct mathematical skills.

Instead of assuming we have many pure items, we begin a longer investigation into pre-requisite discovery with a simplifying assumption that we hope to eventually relax: that the Q-matrix is known. We know of no current method for learning the prerequisite structure among skills in cases where there are very few pure items; so although the method we propose here is limited to cases where the Q-matrix is known, our method solves a novel problem. There are existing techniques for discovering and refining a Q-matrix, so there will be many cases where the Q-matrix is known or can be estimated to some approximation.

In the following sections, we explain the statistical model and the prerequisite discovery procedure. We then describe our evaluation of the procedure on simulated data, where the Q-matrix and the true prerequisite model are known. We conclude by considering our results in the context of educational technology.

2. PREREQUISITE DISCOVERY

As is conventional [4,11,12], we treat "skills/concepts" as latent, or unmeasured variables. Specifically, we model skills as continuous variables that represent the degree to which a student has mastered or has knowledge of a particular skill. We treat items as continuous variables that reflect the degree to which a student completed a task correctly. This idealized conception of an item is rarely even approximated in practice, where the measure of task completion is often a binary variable with values = correct/incorrect. A binary item can, however, be considered as a projection of a continuous item, and correlations among idealized continuous items can be estimated by computing the tetrachoric correlation matrix among the measured binary items.
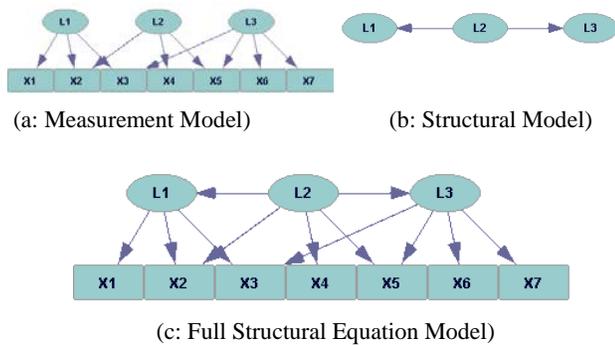

(a: Measurement Model) (b: Structural Model)

(c: Full Structural Equation Model)

Figure 1: Structural Equation Models

The Q-matrix typically defines which items "load" on which latent skills. We can define a "measurement model" that relates latent skills to measured items (Fig 1-a). A prerequisite graph represents what skills must be mastered prior to mastering other

---

[1] Note also that BPC was written for continuous items. It would need to be extended to work on binary data before it could be applied to "correct/incorrect" test items.

skills. If an edge is present and oriented as an arrow from latent L1 to latent L2, then skill L1 is a prerequisite for L2. If edges from two latents L1 and L2 are both causes of a third latent L3, then skills L1 and L2 both influence the degree to which L3 can be mastered. Notably, this conceptualization is distinct from Knowledge Space Theory [4], which represents prerequisite relationships among items, not skills. Instead, it is more closely related to the domain concept map approach [13].

By modeling the relations among the skills as a path analytic causal model among the latent variables (Fig 1-b), called the "structural model," we can then combine the "measurement model" and the "structural model" to form a full *linear structural equation model* [1] (Figure 1 a-c).

By assuming that the measurement model is known, we need not simply specify a structural model representing the prerequisite relations as a causal graph; we can search for it with a causal discovery algorithm [3,2]. The input to a causal discovery algorithm is typically the independence and conditional independence relations that hold among a set of variables, and the output an equivalence class of causal structures that are empirically indistinguishable but consistent with theoretical background knowledge. In our case, however, the variables of interest are latent/unmeasured, so we cannot use this strategy directly. We must find some way, as a pre-processing step, of computing or estimating the independence relations among the latent variables from observable constraints among the measured variables. If we can do so, then the output of a causal discovery algorithm applied to the latent skills will be an equivalence class of plausible pre-requisite relationships that explain observable statistical constraints that hold among the *measured* items.

There are at least two strategies for finding the independence or conditional independence relations among the latent variables needed as input for a pre-requisite discovery algorithm. First, we might simply specify a given measurement model and *estimate* $\phi(\mathbf{L})$, the correlation matrix among the latent variables $\mathbf{L}$. We can then treat the estimated latent correlation matrix $\phi(\mathbf{L})$ *as if* it were a sample correlation matrix among measured variables, and then apply a causal discovery algorithm directly to $\phi(\mathbf{L})$ (Figure 2).
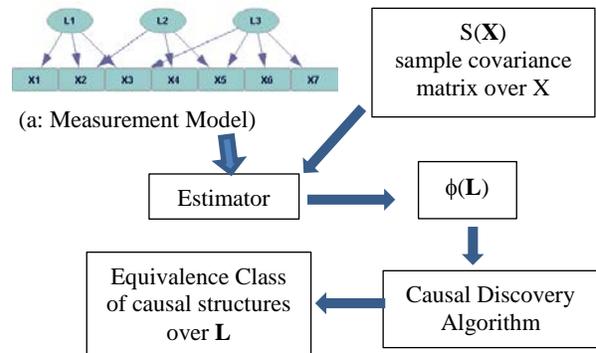

(a: Measurement Model)

**Figure 2: Causal Discovery via Estimating ϕ(L)**

The discovery algorithm would use $\phi(\mathbf{L})$ to make decisions about whether independence and conditional independence constraints over $\mathbf{L}$ hold in the population from which $\phi(\mathbf{L})$ was drawn. The problem with this strategy is that $\phi(\mathbf{L})$ is not a *sample* covariance matrix, it is an *estimate* of a sample covariance matrix. If the sample size for the measured items is N, and the sample covariance matrix among the measured items is S($\mathbf{X}$), then a statistical inference on whether an independence relation holds in

the population from which **S** is drawn is routine, and has a known dependence on the sample size N.[2]

Performing a similar inference from $\phi(\mathbf{L})$ is not routine, as the sample size that should be used is not N, as the covariances in $\phi(\mathbf{L})$ are already estimated from S(**X**) rather than sampled from the same population as **S**. We have no statistical theory about how to correct for N when using an estimate of $\phi(\mathbf{L})$, and the correction is not simple as it depends in a non-trivial way on the measurement model, the number of items relative to the number of skills, and other factors. This lack of statistical theory notwithstanding, the strategy is promising and we intend to investigate its reliability in future work.

A more theoretically satisfactory strategy is as follows: for every conditional independence test among the latents, construct a structural model such that a single coefficient in a full structural equation model vanishes if and only if that particular independence relation among the latent variables holds. If the remainder of the structural model imposes no additional constraints on the measured variables, and the measurement model is known to be correct, then the only thing we need to put to a test is whether the edge coefficient in question is in fact 0. In a structural equation model, the confidence interval around a single parameter estimate is conditional on the other parts of the model being correctly specified. In this case, we hypothesize that the measurement model is correctly specified, and construct the structural model so that no constraints are entailed except the constraint that would hold were the edge in question to be absent.

It turns out that we can fix the measurement model as given, and construct a structural model for each test of independence we desire. For example, consider the edge labelled β in the structural models shown in **Figure 3** (a) and (b).

<center>(a)   (b)</center>
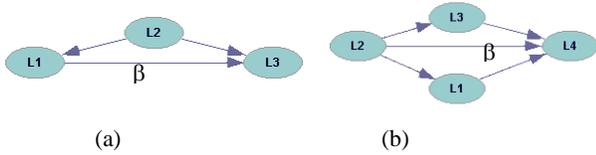
Figure 3: Structural Models to test independence relations

In **Figure 3** (a), $\beta = 0$ if and only if $L_1 \perp\!\!\!\perp L_3 \mid L_2$.[3] In **Figure 3** (b), $\beta = 0$ if and only if $L_2 \perp\!\!\!\perp L_4 \mid L_1, L_3$. If we attach measurement models to these structural models, and estimate the resulting full structural equation models, then the Fisher Information matrix of the coefficient estimates provides asymptotically correct standard errors (Bollen, 89). We can thus use an asymptotically correct statistical inference on β as a surrogate for an asymptotically correct test of $L_1 \perp\!\!\!\perp L_3 \mid L_2$ in **Figure 3** (a), and as an asymptotically correct test of $L_2 \perp\!\!\!\perp L_4 \mid L_1, L_3$. in **Figure 3** (b), with no sample size correction needed.

The disadvantage of this strategy is that one must know, ahead of time, how to construct a structural model to correctly perform any independence test, and then execute a series of such tests by

estimating a series of different structural equation models. Both of these problems can be solved.

If the original set of latent variables is **L**, and the independence relationship under test is: $L_i \perp\!\!\!\perp L_j \mid \mathbf{L_c}$, where L_i, L_j, $\mathbf{L_c}$ and the remaining latents $\mathbf{L_r}$ exhaust and partition **L**, then the structural model to test : $L_i \perp\!\!\!\perp L_j \mid \mathbf{L_c}$ can be built as follows: [4]

Algorithm CITSEMB
(Conditional Independence Test SEM Builder):

1) Form an arbitrary ordering over the variables in $\mathbf{L_c}$, and an arbitrary ordering over the variables in $\mathbf{L_r}$, such that every variable in $\mathbf{L_c}$ is prior in the order to every variable in $\mathbf{L_r}$. Add an edge in the structural model from every variable in $\mathbf{L_c} \cup \mathbf{L_r}$ to every later variable in $\mathbf{L_c} \cup \mathbf{L_r}$.
2) Add an edge from L_i to every variable in $\mathbf{L_c} \cup \mathbf{L_r}$.
3) Add an edge from every variable in $\mathbf{L_c}$ to L_j
4) Add an edge from L_j to every variable in $\mathbf{L_r}$
5) Add an edge from $L_i \rightarrow L_j$, with coefficient b.

For example, if $\mathbf{L} = \{ L_i, L_j, L_k, L_l, L_m\}$, and $\mathbf{L_c} = \{L_k, L_l\}$, and $\mathbf{L_r} = \{L_m\}$, then the structural model to construct in order to test $L_i \perp\!\!\!\perp L_j \mid L_k, L_l$ is shown in Fig. 4.
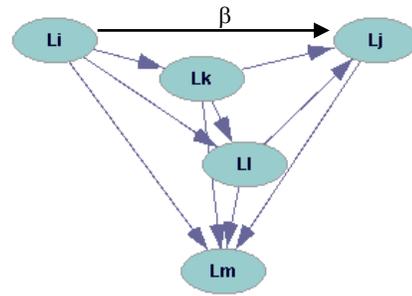
Figure 4

If, when a measurement model is attached and a full structural equation model estimated, the null hypothesis that $\beta = 0$ corresponds to the hypothesis: $L_i \perp\!\!\!\perp L_j \mid L_k, L_l$, and no other independence constraint among the latent variables is entailed by the model.

Claim: the structural equation model constructed as specified in section 2 entails $L_i \perp\!\!\!\perp L_j \mid \mathbf{L_c}$ and no other constraints, just in case $\beta = 0$.

Proof. There are no constraints among the variables in $\mathbf{L_c} \cup \mathbf{L_r}$, as the variables in this set are connected with a complete graph. Consider the graph **G** that results from removing the edge $L_i \rightarrow L_j$ (equivalently, set $\beta = 0$). In **G**, $L_i \perp\!\!\!\perp L_j \mid \mathbf{L_c}$, as every trek

---

[2] By assuming that X is multivariate normal, we can use tests for vanishing correlations and partial correlations as tests for independence and conditional independence.

[3] The notation $A \perp\!\!\!\perp B \mid C$ means that A is probabilistically independent of B given C, i.e., $P(A \mid C) = P(A \mid B,C)$, for all values a of A, b of B, and c of C.

[4] This is an extension of a similar method suggested in [3, chapter 12]. In that algorithm, the measurement model was assumed to be correct, and also "pure," i.e., no cross factor loadings or correlated errors. As a result of it being "pure", the structural model could be formed using the subset of **L** involving only L_i, L_j, and $\mathbf{L_c}$, with the latent variables in $\mathbf{L_r}$ and their accompanying items left out of the model. In our context, the measurement model will never be pure – by instructional design. As a result, we must include all the variables in L in the technique we discuss.

between L_i and L_j passes through $\mathbf{L_c}$, and no variable in $\mathbf{L_c}$ is a collider on any undirected path from L_i to L_j that does not also involve a non-collider prior in the order within $\mathbf{L_c}$. No proper subset of $\mathbf{L_c}$ d-separates L_i and L_j, as there is a path from $L_i \rightarrow V \rightarrow L_j$ for every variable $V \in \mathbf{L_c}$. No variable in $\mathbf{L_r}$ is in any set that d-separates L_i and L_j, because each variable V in $\mathbf{L_r}$ is a collider on an undirected path between L_i and L_j involving only V. Thus the only independence constraint entailed by $\mathbf{G}$ is: $L_i \perp\!\!\!\perp L_j \mid \mathbf{L_c}$. Q.E.D.

We use this method to perform independence and conditional independence tests as required by the PC algorithm. [3]

3. VALIDATION ON SIMULATED DATA
3.1 Methods
Our goal was to measure the method's ability to recover prerequisite structure when we varied (i) the structural model, (ii) the purity of the measurement model, (iii) the sample size, and (iv) whether the observed data was continuous or binary. In each of these conditions we performed 100 simulations with different parameterizations.

We used three structural models representing different causal relations between the latent skills. Note, however, that it is impossible to recover the entire structure using constraint-based search; instead we can only recover the Markov equivalence class that each structure belongs to. Therefore, the direction of some edges cannot be learned from data. The three equivalence classes of our structural models are shown in **Figure 5**, a-c. These three figures represent the most we can hope to learn about the structural models from data; note the undirected edges.
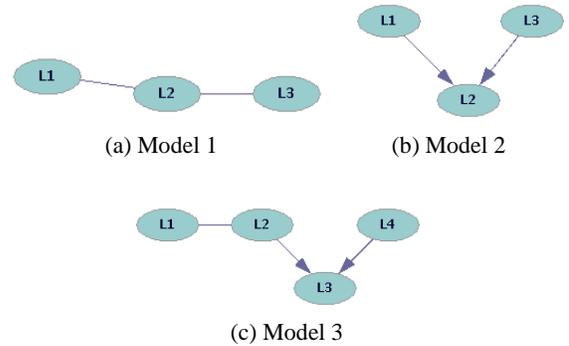

(a) Model 1   (b) Model 2

(c) Model 3

Figure 5: Equivalence classes of generating structural models
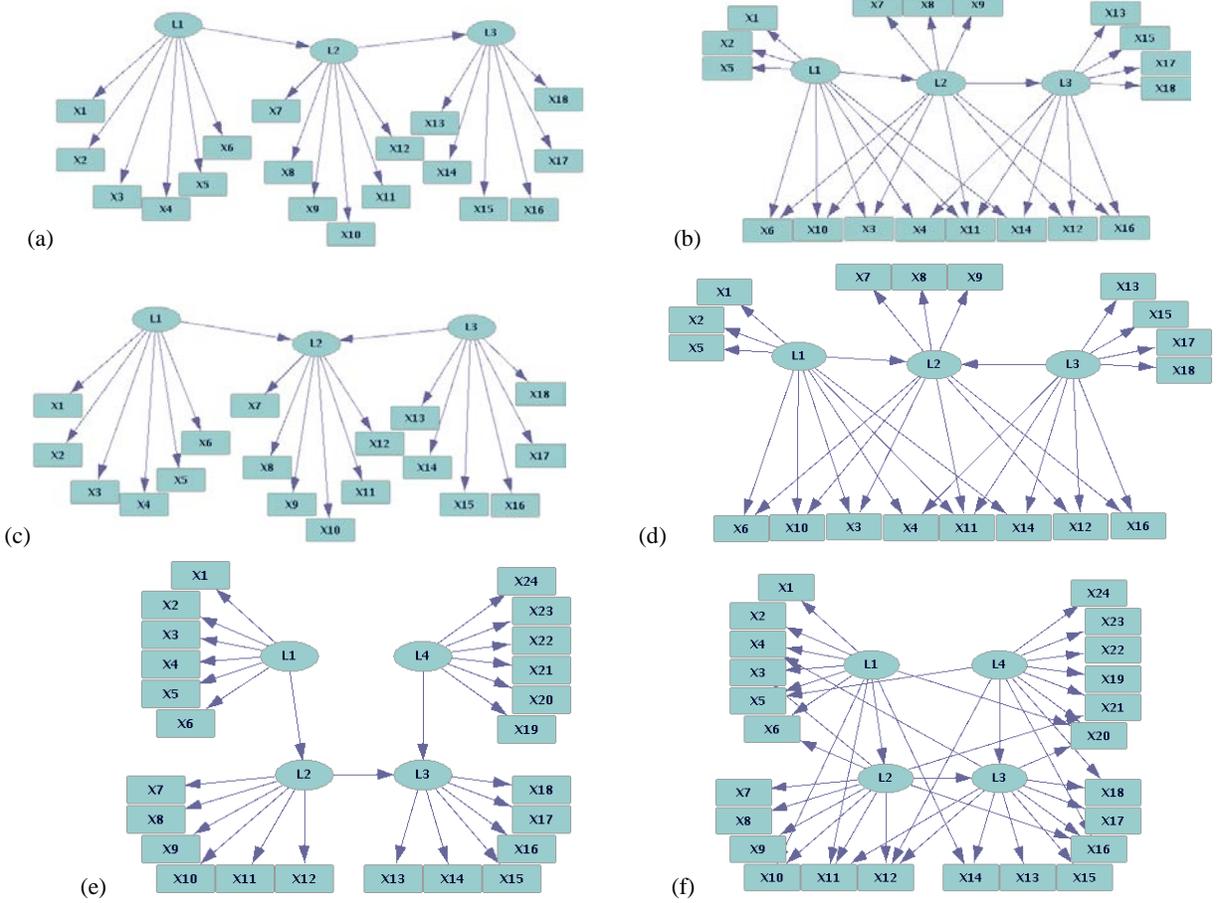

Figure 6: Full models used to generate simulation data. Pairs with the same structural models but different measurement models (pure v. impure) are arranged next to each other. The equivalence class of the structural models for (a) and (b) is shown in Figure 5 (a); (c) and (d) correspond to Figure 5 (b); and (e) and (f) correspond to Figure 5 (c).

We simulated data from these three different structural models, using either a pure or an impure measurement model, for a total of six different generating models (Figure 6, a-f). The simulated data represent student responses to items that load on the skills.

For each generating model, we instantiated the model 100 times, each with new parameters chosen to be positive (so that skills would have a positive effect on item responses) and large enough to care about. Coefficient were drawn uniformly from [.5, 1.5], covariances drawn from [.2, .3], and error variances drawn from [1.0, 3.0]. For each instantiation we simulated two datasets, one of size n=150 and one with n=1000. This gave us 1200 unique datasets.

For each dataset, we created a binary copy, to represent the conventional case that student responses are scored as binary (correct / incorrect) rather than continuously. For each measurement j of each observed variable X_i, the binary projection of that measurement, X_{ij}^{Bin}, was set to zero if $X_{ij} <$ mean(X_i), and 1 otherwise, to avoid choosing multiple arbitrary cutoff points. We then used the function `hetcor()` in the R package `Polycor` to estimate the correlation matrix for each truncated dataset.

We used the `lavaan` R package to estimate all structural equation models.[15] We created a simple function, `semTest()`, which takes (a) a measurement model, (b) a dataset (or covariance matrix & sample size), (c) a pair of variables L_i and L_j, and (d) a conditioning set $\mathbf{L_c}$. `semTest` then builds the correct `lavaan`-syntax model to test $L_i \perp\!\!\!\perp L_j \mid \mathbf{L_c}$ (according to the CITSEMB algorithm described above), estimates the model using the `sem()` function in the `lavaan` package, and returns the *p*-value of the critical edge.

We used `semTest()` as the independence test in the PC algorithm, as implemented in the R package `pcalg`. [16] For each of the 1200 continuous datasets, and each of the 1200 estimated covariance matrices of the binary datasets, we ran the PC algorithm (using an alpha value of 0.02) and produced an equivalence class for the structural model in which we assumed no additional latent confounding, called a *pattern* [14].

We then scored each graph on the following metrics:

1. *True positive adjacencies* or *adjacency recall* (# correct adjacencies in output / # adjacencies in true graph),
2. *False positive adjacencies* (# incorrect adjacencies in output / # gaps in true graph),
3. *True adjacency discovery rate* or *adjacency precision* (# correct adjacencies in output / # adjacencies in output)
4. *True positive orientations* or *orientation recall* (# correctly oriented edges in output / # orientable edges in true equivalence class). Defined to be 1 if none of the edges in the true equivalence class are orientable, as is the case for Model 1 (see **Figure 5** (a)).
5. *False positive orientations* (# incorrectly oriented edges in output / # edges in true equivalence class) There are two ways to incorrectly orient an edge: reverse the true orientation, or orient an edge that is undirected in the true equivalence class.
6. *True orientation discovery rate* or *orientation precision* (# correctly oriented edges in output / # oriented edges in output). Defined to be 1 if none of the edges in the output are oriented.
7. *False negative orientation rate* (# incorrectly unoriented edges in output / # oriented edges in true equivalence class).

3.2 Results

Our results show that the algorithm performs well for discovering adjacencies (Figure 7). Even in the most difficult (and most realistic) case, where the sample size is 150, the measurement model is impure, and the data is binary, we still recover 74% of adjacencies for Model 1, 76% for Model 3, and 89.5% for Model 2.
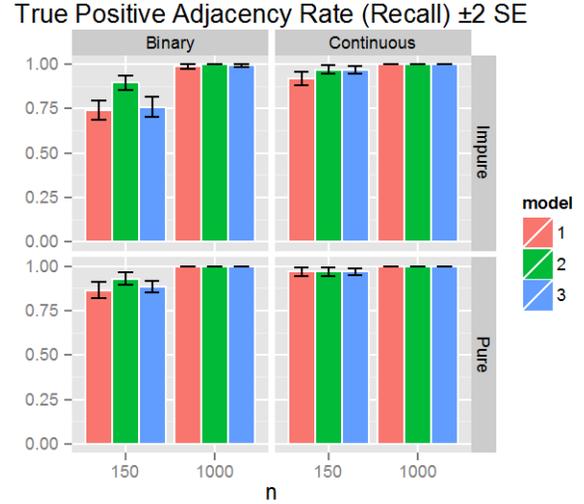
Figure 7: True positive adjacency rates (i.e. # correct adjacencies in output / # adjacencies in true graph), ± two standard errors. Grouped by the three structural models (see Figure 5), sample size of 150 vs. 1000, pure vs. impure measurement models, and continuous v. binary data.

The false positive rate for adjacencies (Figure 8) is extremely low in all conditions.
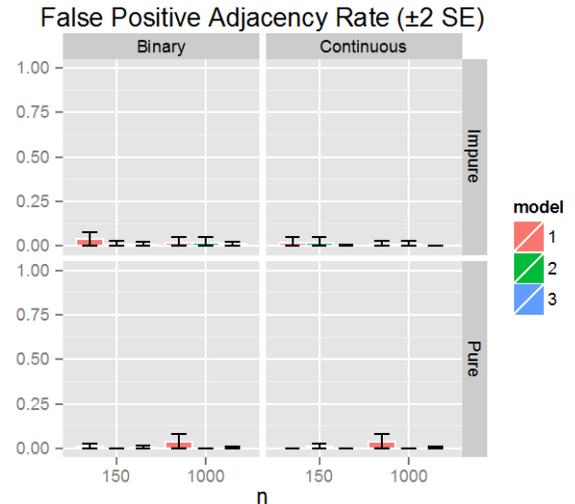
Figure 8: False positive adjacency rates (i.e. # incorrect adjacencies in output / # gaps in true graph), ± two standard errors. Grouped by the three structural models (see Figure 5), sample size of 150 vs. 1000, pure vs. impure measurement models, and continuous v. binary data.

The true discovery rate for adjacencies, or *precision*, is near perfect in all conditions (Figure 9).

5

True Discovery Adjacency Rate (Precision) ±2 SE
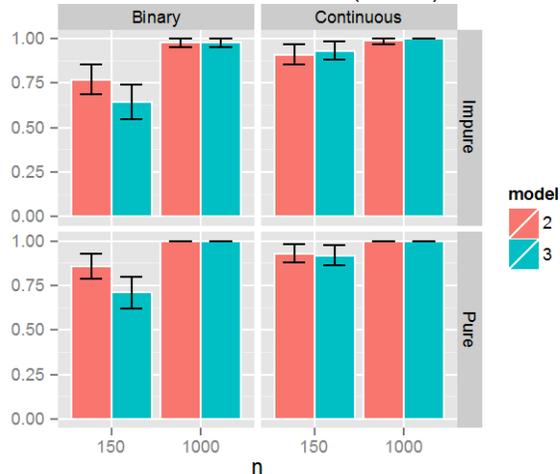

True Positive Orientation Rate (Recall) ±2 SE
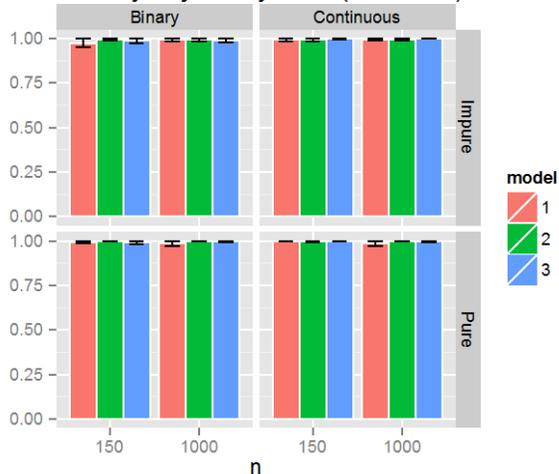
Figure 9: True Discovery Rates for adjacencies (precision) (i.e. # correct adjacencies in output / # adjacencies in output), ± two standard errors. Grouped by the three structural models (see Figure 5), sample size of 150 vs. 1000, pure vs. impure measurement models, and continuous v. binary data.

The algorithm does not perform quite as well for edge orientations, for two reasons. First, an edge can only be oriented if the adjacency it orients has been discovered. Second, adjacency errors in nearby edges ramify to produce additional orientation errors, because the orientation decisions involve interaction between nearby edges. As a result, if the algorithm makes adjacency errors, these will typically produce orientation errors. So in this sense orientation is harder than adjacency.
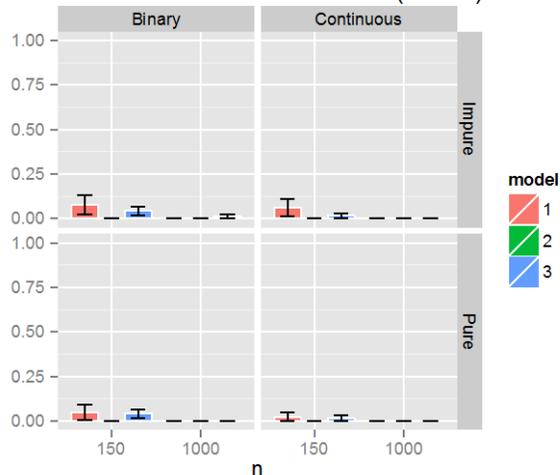
The true positive orientation rate (recall) is shown in Figure 10; the worst score is 64.5% (for Model 3, with binary data, an impure measurement model and sample of 150). Note however that many of these errors are caused by missing adjacencies (it is impossible to orient an edge that is not discovered). Compare with the False Negative Rate (Figure 13), which is relatively low, indicating that of the orientable adjacencies that were recovered, most of them were correctly oriented.

Figure 10 True positive orientation rates (orientation recall) (# correctly oriented edges in output / # orientable edges in true equivalence class; defined to be 1 if none of the edges in the true equivalence class are orientable, which is true for Model 1, so Model 1 is not shown.)

False Positive Orientation Rate (±2 SE)

Figure 11: False positive orientation rates (i.e. # incorrectly oriented edges in output / # edges in true equivalence class). There are two ways to incorrectly orient an edge: reverse the true orientation, or orient an edge that is undirected in the true equivalence class.

6

The algorithm performs very well with regard to false positives (Figure 11) and precision (Figure 12), and quite well for false negatives (Figure 13).
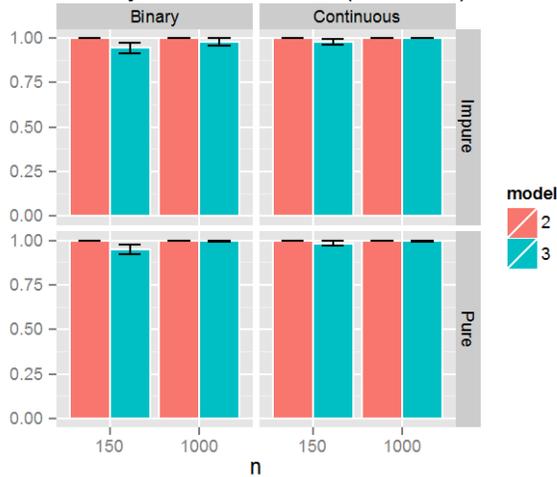

Figure 12: True discovery orientation rate or precision (i.e. # correctly oriented edges in output / # oriented edges in output). Model 1 is omitted because there are no orientable edges in the equivalence class, so an oriented edge in the output is guaranteed to be incorrect, making the false positive rate a better measure of performance.
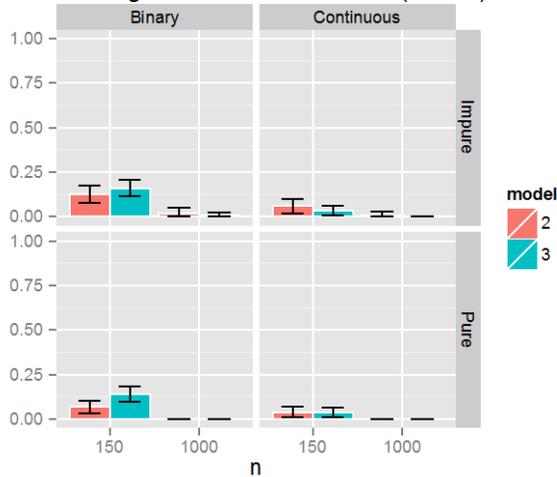

Figure 13: False negative orientation rate (# incorrectly unoriented edges in output / # oriented edges in true equivalence class). Model 1 is omitted because there are no oriented edges in the true equivalence class.

4. CONCLUSIONS

The prerequisite graph is an important pedagogical artifact in itself, because we can use it to examine the structure of a domain, and it is furthermore a critical element of adaptive learning environments, where it can be used to create personalized and efficient learning trajectories for students. We expect that our

algorithm can be used to discover fine-grained prerequisite structures to make student learning more efficient and more effective. There are multiple ways that this may be the case. For instance, an accurate prerequisite graph can be used to eliminate implausible curricular sequences. It can also be used as a basis for interleaving and spacing strategies to enable robust learning.

One contribution of this work is that we can infer prerequisite structure without relying on variation in topic ordering in existing data. While sometimes data with varied topic orderings are naturally available [10], in general, collecting such data requires experimentation. Specifically, it requires offering alternative topic sequences to different students, which necessarily means that some students are exposed to a suboptimal curricular sequence. By inferring prerequisite structure based on a single assessment, we avoid having to waste student time and effort.

Moreover, by positing an entire structural equation model relating all skills and all items, the algorithm developed here is an advance on testing pairwise prerequisite relationships, as in [10].

Evaluation of prerequisite structure is non-trivial, as [10] note. There are no established evaluation procedures, metrics, or standard data sets. Evaluation on simulated data provides an important step forward, because simulated data allow us to evaluate performance objectively. In other words, with simulated data we know the true state of the world, and we can measure whether our models are able to recover the true state.

Our algorithm is the only method currently available for inferring latent structure when the measurement model contains few pure items (i.e. items that load on only one latent). It performed well in our simulation.

There are several limitations, however: first, the simulation may fail to represent the variability in real-world data. Second, the simulation may fail to represent the complexity or form of latent skill structures in the real world. We assume an underlying linear model, which fails to capture the fact that prerequisite relationships are often interactive – learning more of one prerequisite cannot compensate for the lack of another. We also assume no confounding by latent variables other than our knowledge components. We would like to check how robust our method is to violations of these assumptions. Lastly, the method as it stands is unable to handle cases where there are few items per latent skill, because in those cases the structural equation models we use for testing conditional independences are under-identified. We intend to tweak the CITSEMB algorithm to handle these underidentified cases.

While our evaluation covered a range of plausible variable values and considered a number of generating use cases, we intend to extend this work with an evaluation on real-world data in future work. We also intend to extend the work by investigating the robustness of the procedure to errors in the Q-matrix specified. It is worth pointing out, however, that Q-matrix discovery algorithms could be combined with prerequisite discovery algorithms of the type we present.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1]   Bollen, K. (1989). *Structural Equation Models with Latent Variables*. Wiley

[2] Rau, M., Scheines, R., Aleven, V., and Rummel, N. (2013). Does Representational Understanding Enhance Fluency – or Vice Versa? Searching for Mediation Models. *Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013)*.

[3] Spirtes, P, Glymour, G., Scheines, R. (2000). *Causation, Prediction and Search*, 2nd Edition, MIT Press.

[4] Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., & Johannesen, L. (1990). Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review*, 97(2), 201.

[5] Tatsuoka, K. K. (2012). *Cognitive Assessment: An Introduction to the Rule Space Method*. Routledge.

[6] Nathan, M. J., & Koedinger, K. R. (2000). An Investigation of Teachers' Beliefs of Students' Algebra Development. *Cognition and Instruction*, 18(2), 209–237.

[7] Barnes, T., Bitzer, D., & Vouk, M. (2005). Experimental Analysis of the Q-Matrix Method in Knowledge Discovery. In M.-S. Hacid, N. V. Murray, Z. W. Raś, & S. Tsumoto (Eds.), *Foundations of Intelligent Systems* (pp. 603–611). Springer Berlin Heidelberg.

[8] González-Brenes, J. P., & Mostow, J. (2013). What and When do Students Learn? Fully Data-Driven Joint Estimation of Cognitive and Student Models. In *Proceedings of 6th International Conference on Educational Data Mining*. Memphis, TN.

[9] Liu, J., Xu, G., & Ying, Z. (2012). Data-Driven Learning of Q-Matrix. *Applied Psychological Measurement*, 36(7), 548–564.

[10] Vuong, A., Nixon, T., & Towle, B. (2011). A Method for Finding Prerequisites Within a Curriculum. In *Proceedings of 4th International Conference on Educational Data Mining* (pp. 211–216).

[11] Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4), 253–278.

[12] Kupp, A. A., Templin, J., & Henson, R. A. (2012). *Diagnostic Measurement: Theory, Methods, and Applications*. Guilford Press.

[13] Heller, J., Steiner, C., Hockemeyer, C., Albert, D., Heller, J., Steiner, C., … Albert, D. (2006). Competence-Based Knowledge Structures for Personalised Learning. *International Journal on E-Learning*, 5(1), 75–88

[14] Pearl, J. (2009). *Causality: Models and Reasoning, and Inference*, 2nd Edition. Cambridge University Press

[15] Rosseel, Y. (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, *48*(2), 1–36.

[16] Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., & Bühlmann, P. (2012). Causal Inference Using Graphical Models with the R Package pcalg. *Journal of Statistical Software*, *47*(11), 1–26.

[17] Silva, R. Scheines, R., Glymour, C., Spirtes, P. (2006). "Learning the structure of linear latent variable models."*The Journal of Machine Learning Research* 7: 191-246.