Joint Discovery of Skill Prerequisite Graphs and Student Models

Yetian Chen[1] José P. González-Brenes[2] **Jin Tian**[1]

[1]Computer Science Department, Iowa State University, Ames, IA, US

[2]Advance Computing and Data Science Lab, Pearson, San Diego, CA, USA

June 28, 2016

Table of contents

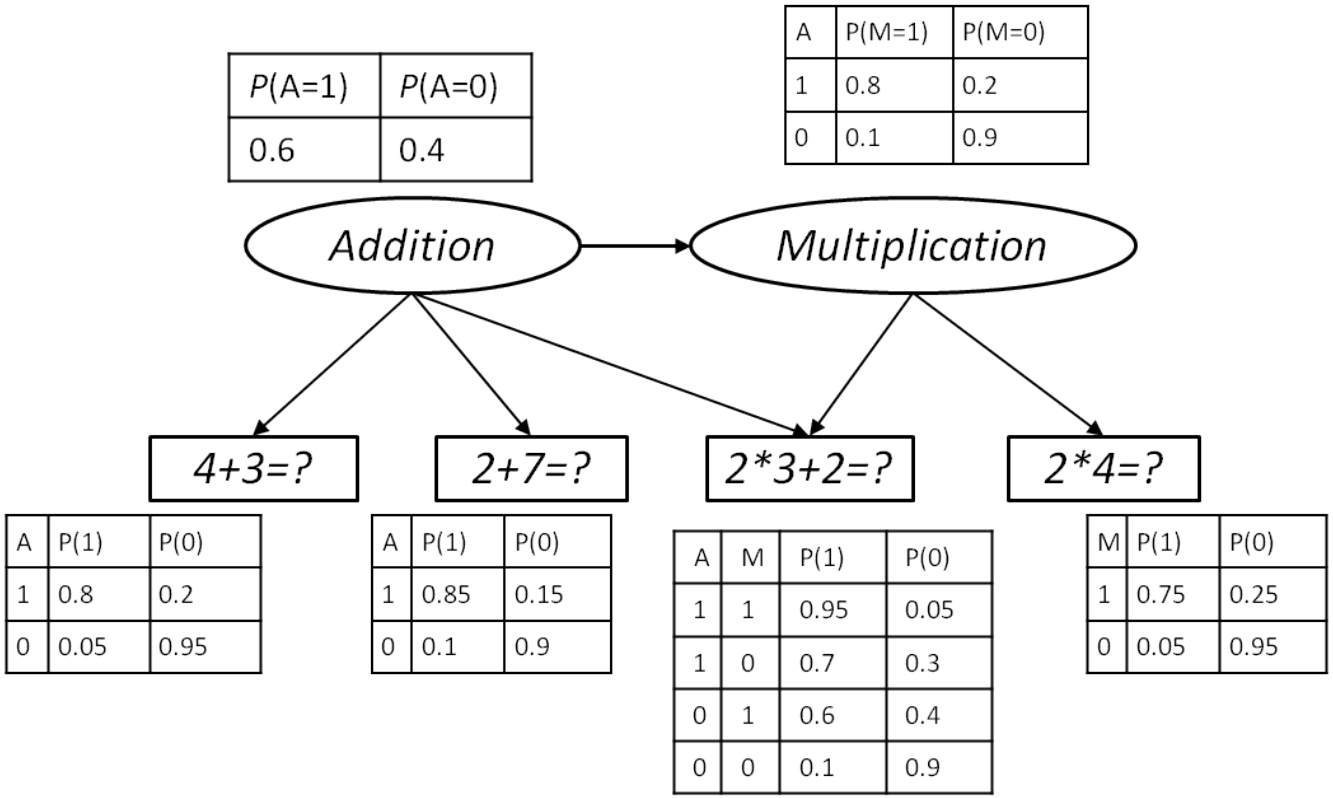
Knowledge graphs

Knowledge Graphs (a.k.a prerequisite networks):

- Are useful for Intelligent Tutoring Systems that assess student knowledge or that provide remediation to learners
- Model the prerequisite relationships between skills and the dependencies between skills and items
- Bayesian networks can represent Knowledge Graphs

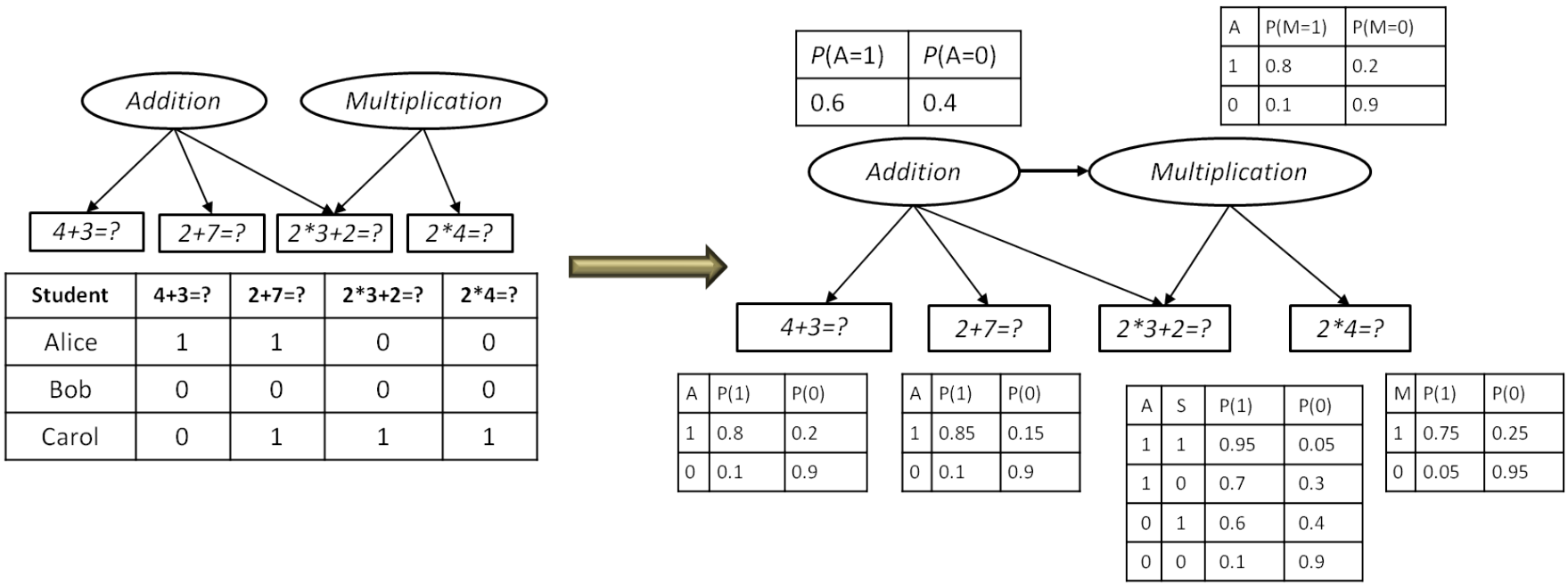
Bayesian Networks

Bayesian networks provide a compact representation of the joint distribution over skill and item variables.



Prerequisite Discovery as a Machine Learning Problem

- Student performance data (what items a learner answers correctly)
- Skill-to-item mapping Q-matrix is known
- Student's mastery of a skill is unknown $\rightarrow$ latent variables
- Learning Bayesian networks with latent variables



Related Prior Work

- Brunskill (2010) and Chen et al. (2015)'s work:
  - estimated only the pairwise relationships, unable to tell if the relationships are due to indirect (e.g, $S_3 \rightarrow S_2 \rightarrow S_1$), or direct (e.g, $S_3 \rightarrow S_2 \rightarrow S_1$) effects.
  - unclear how to use the output of these relationships for student modeling
- Käser et al. (2014): manually specified the Bayesian network structure
- Scheines et al. (2014): learned the prerequisite graph as a Bayesian network but did not address the issue of Markov equivalence between Bayesian networks.

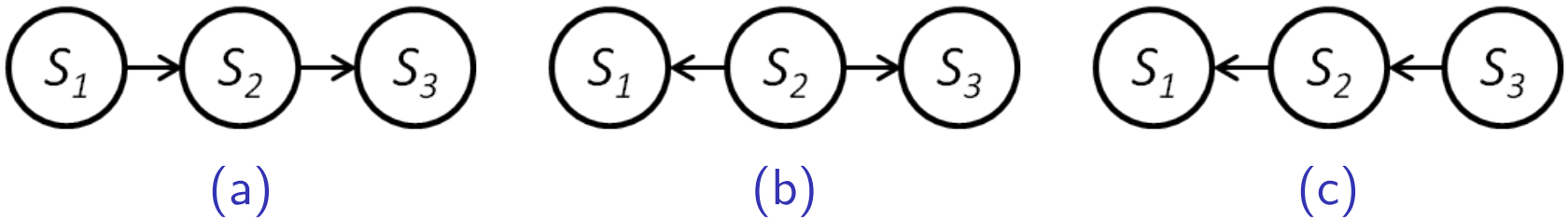
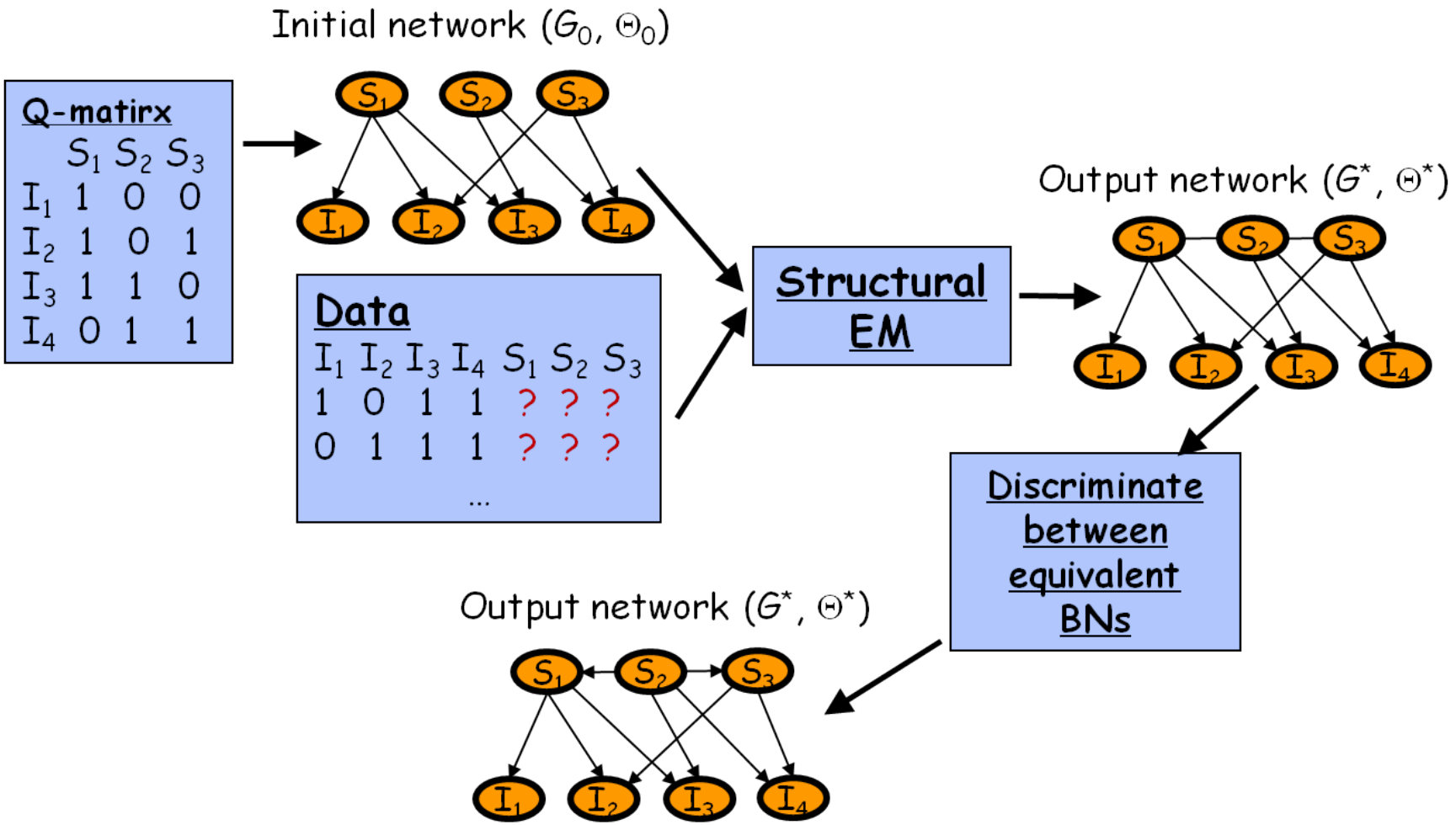
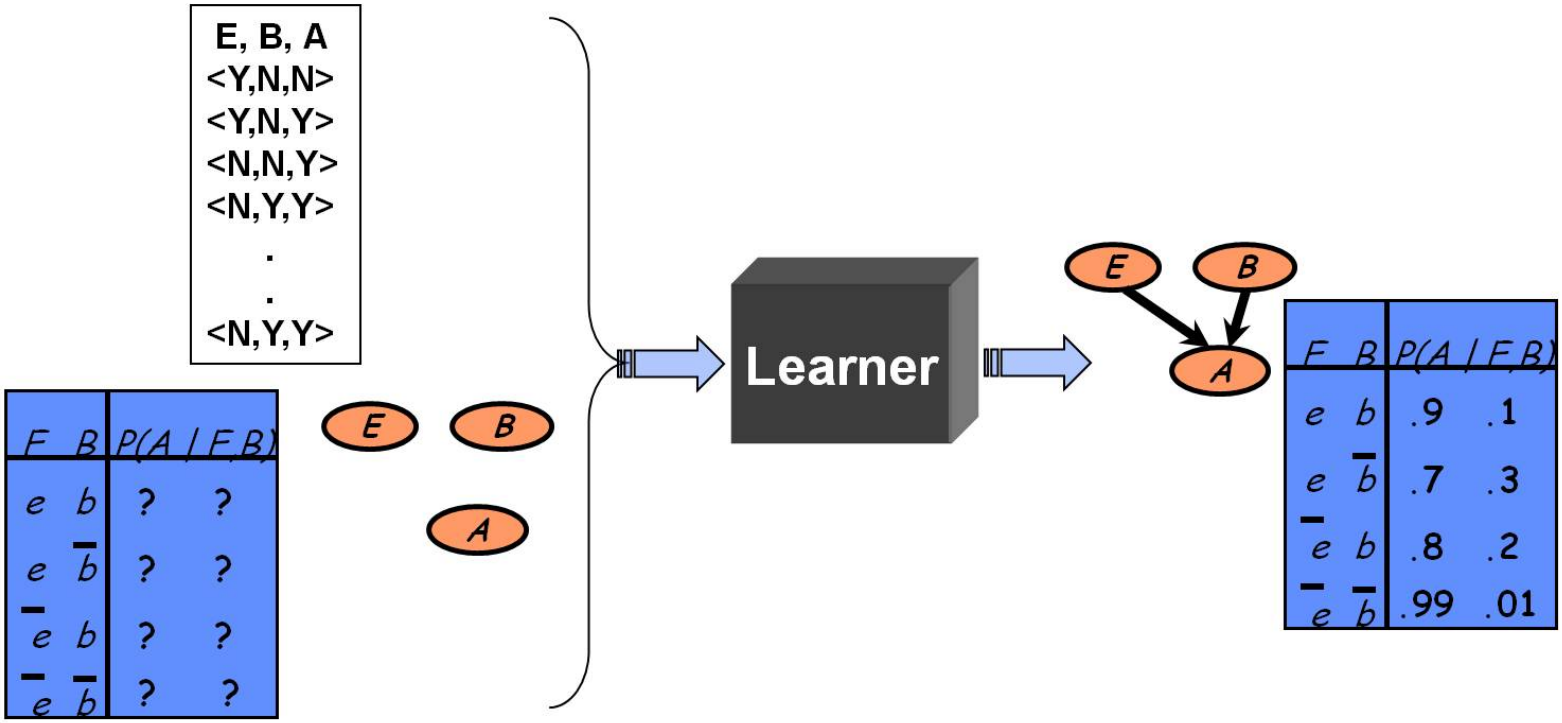


Figure: Three equivalent BNs representing different prerequisite structures.

The COMMAND (Combined student Modeling and prerequisite Discovery) Algorithm



Learning Bayesian Networks from Data

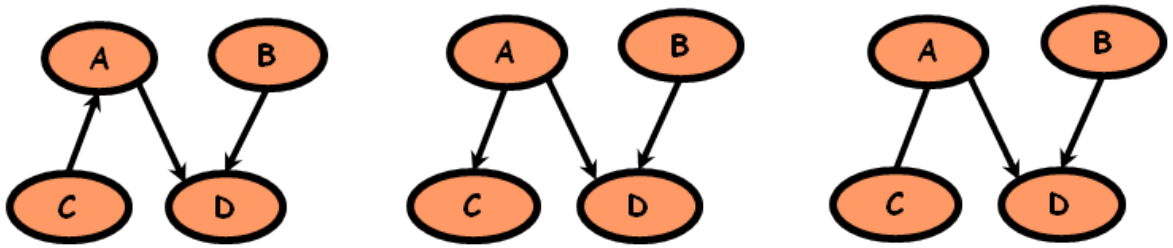


Two phases:

1. Construct the topology (structure) of the networks.
2. Estimate the parameters of the CPDs given the fixed structure.

What Can We Learn?

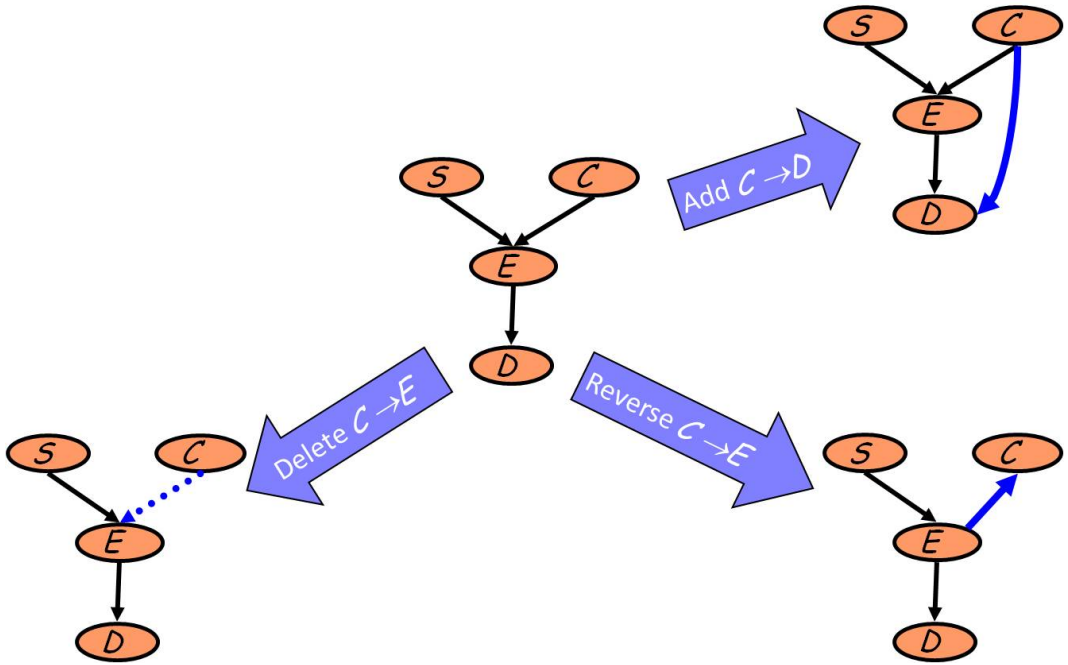
1. Assumption: there exists a BN B that perfectly represents $P(\mathbf{X})$.
2. Two BNs are Markov equivalent if they represent the same set of CIs.
3. Markov equivalent BNs are statistically indistinguishable given only observational data.
4. All Markov equivalent BNs belong to the same equivalence class G^* that can be represented by a unique complete partially DAG (CPDAG).



Two Markov equivalent DAGs CPDAG

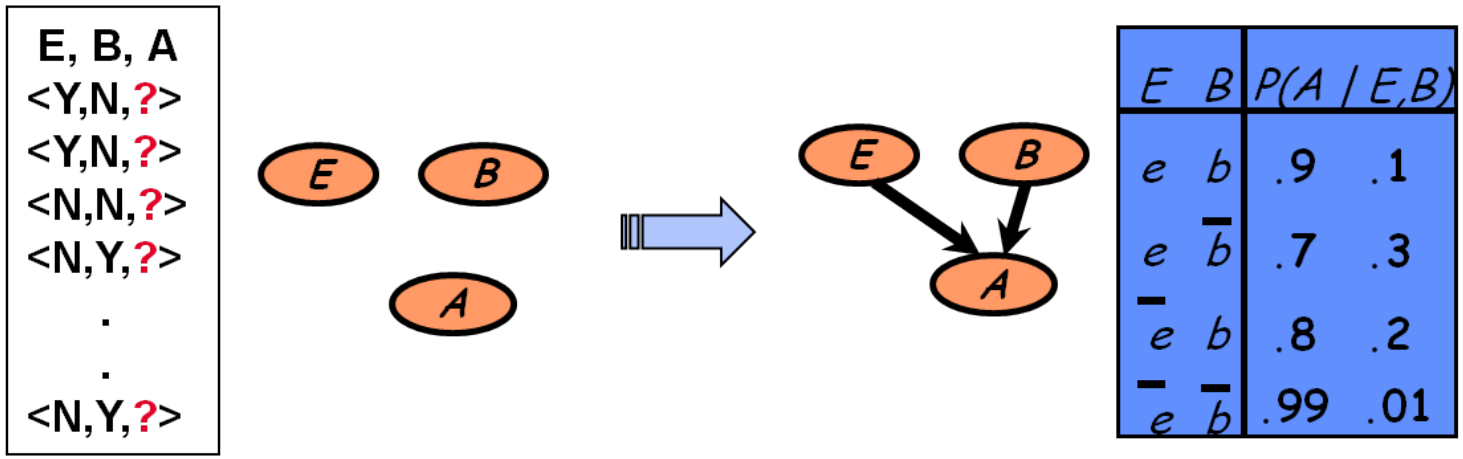
Score-based Search

- Score-based search:
  - Solve an optimization problem.
  - A score to measure the fitness of a DAG to the data: $Score(G : D) = \log P(G, D)$
  - Maximize the score by searching in the space of possible DAGs.
  - Local search, e.g., greedy hill climbing, simulated annealing, etc.



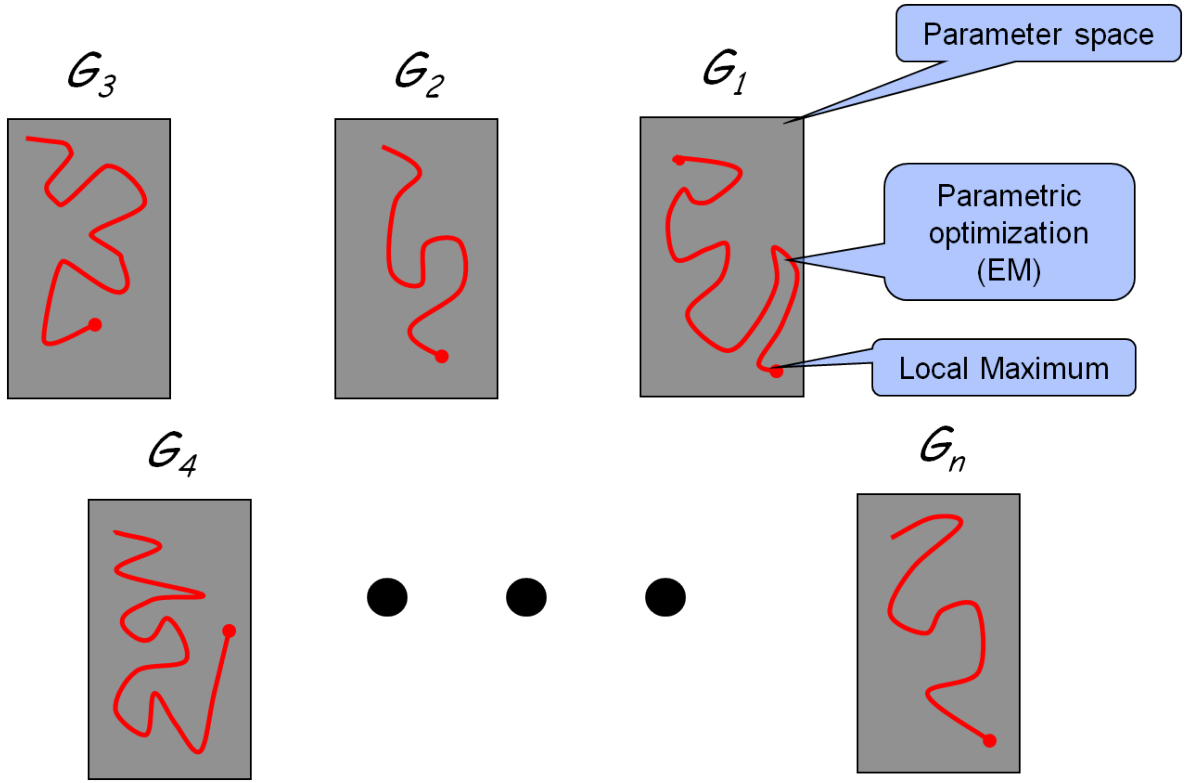
What If Some Variables Are Not Observed

- Data contains latent variables or missing values.



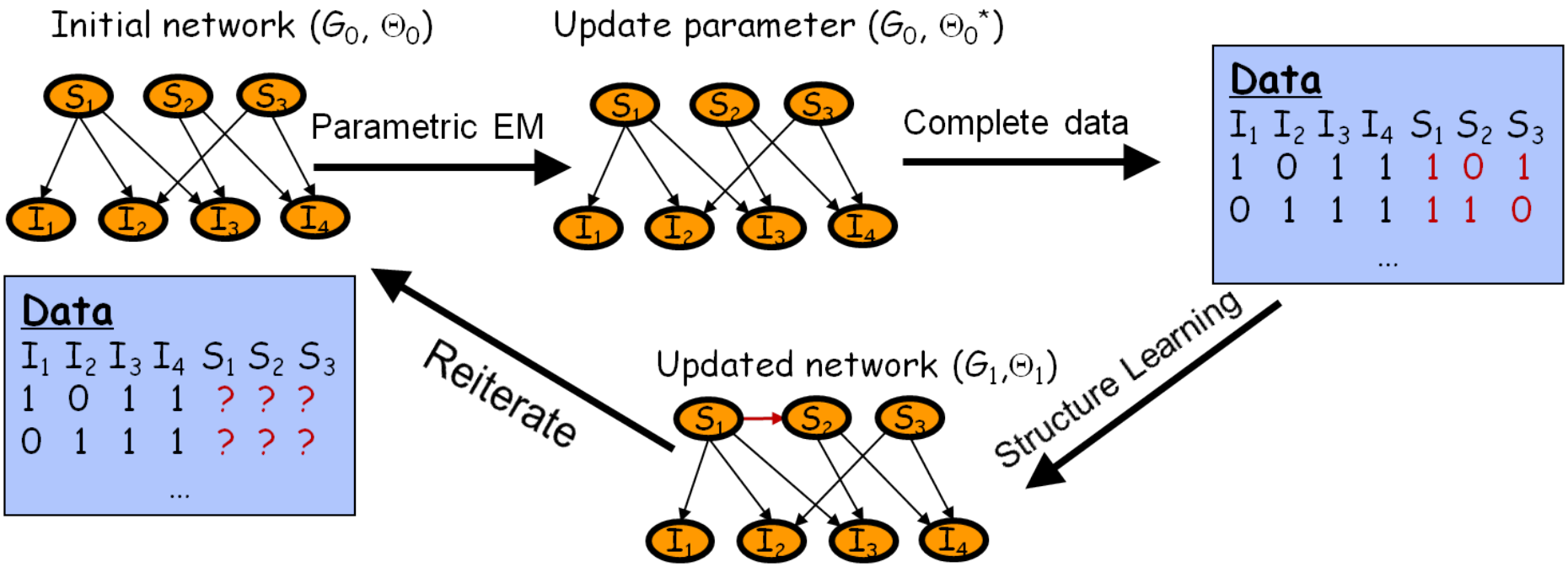
- $Score(G : D)$ has NO closed form solution.
- Use iterative algorithm, mostly EM, to estimate $Score(G : D)$.

Traditional EM



- Need run EM for each candidate BN structure.
- EM requires BN inference, once for EACH iteration.
- EM takes a large (hundreds) number of iterations to converge.
- Computationally prohibitive.

Structural Expectation Maximization (Structural EM)



(Friedman et al, 1997;1998)

Advantage:

- We run EM only on ONE structure, namely the current structure.

Structural EM: Convergence Property

- $Score(G_t, \theta_t : D)$ increases monotonically with iteration t. Hence structural EM converges.
- θ_t converges to global or local parametric maxima or saddle points in the parameter space.
- Not sure the structure converges to what.
- Empirical results indicates that structural EM finds good structures.

Discriminate Between Equivalent BNs

- Scoring function used by Structural EM does not discriminate between equivalent Bayesian networks.

(a) (b) (c)

Figure: Three equivalent BNs

- We need determine the orientation for every reversible edge.

A Heuristic to Orient A Reversible Edge

Assumption

If S_1 is a prerequisite of S_2 (i.e., $S_1 \to S_2$), then $S_1 = 0 \Rightarrow S_2 = 0$. In other words, $P(S_2 = 0|S_1 = 0) = 1$.

Rule

For every reversible edge, if $ratio = \frac{P(S_2=0|S_1=0)}{P(S_1=0|S_2=0)} \geq 1$, we determine $S_1 \to S_2$; otherwise, we determine $S_1 \leftarrow S_2$.

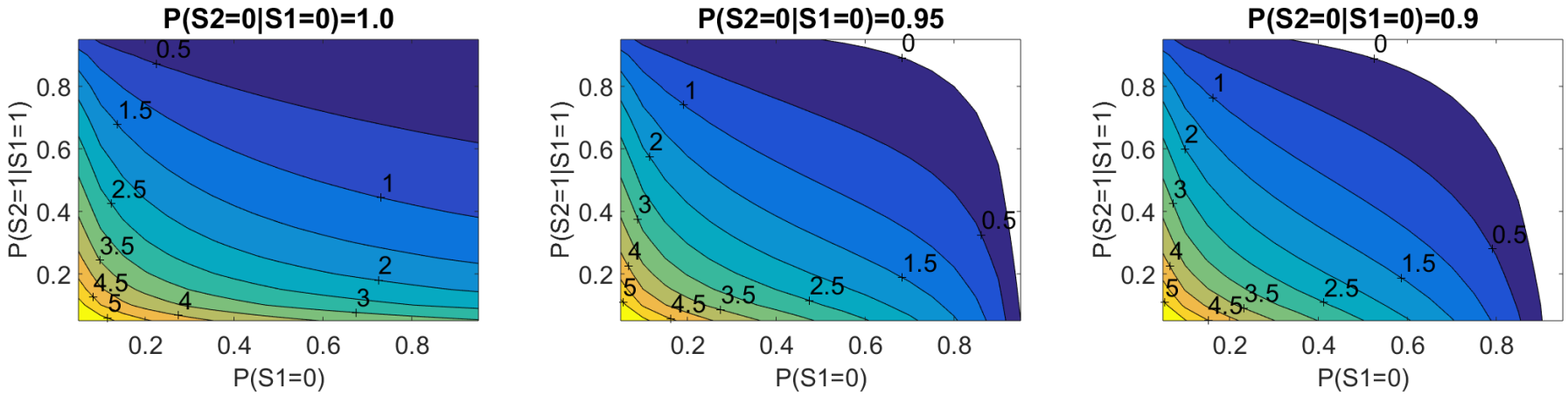
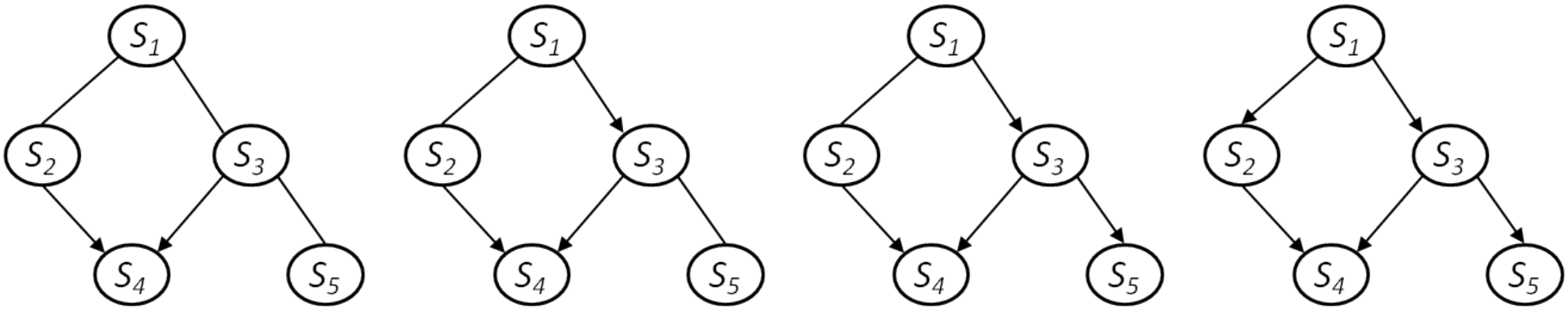


Figure: Contour plots of $log(ratio)$ against $P(S_1 = 0)$ and $P(S_2 = 1|S_1 = 1)$ for various values of $P(S_2 = 0|S_1 = 0)$

An Ad-hoc Strategy to Orient All Reversible Edges

1. For each reversible edge $S_i - S_j$, let $ratio^* = ratio$ if $ratio \geq 1$ and $ratio^* = \frac{1}{ratio}$ otherwise.
2. Sort the list of reversible edges by $ratio^*$ in descending order.
3. Orient the edges by this ordering using the heuristic rule.
4. After determine each edge, propagate the constraint to mantain equivalence and acyclicity.

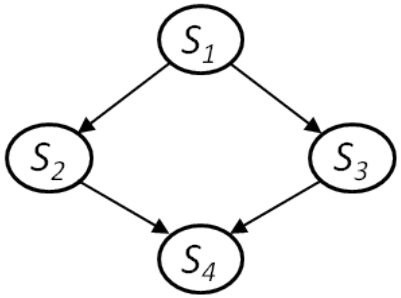


(a) 3 reversible edges (b) Orient $S_1 \rightarrow S_3$. (c) Force $S_3 \rightarrow S_5$ (d) Orient $S_1 \rightarrow S_2$.

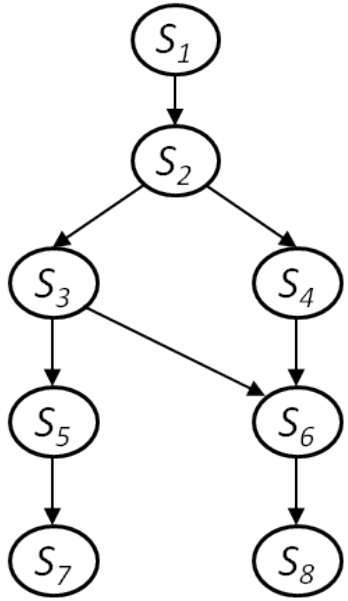
Evaluation: Synthetic Skill Prerequisite Graph

- Each skill node is parent of 6 item variables and each item variable has 1-3 skill nodes as parents.
- All of these nodes are modeled using binary random variables. Skill node: mastery or not mastery; item node: correct or incorrect

(a) Structure 1



(b) Structure 2



(c) Structure 3

Figure: Item nodes are omitted.

Evaluation With Synthetic Data

We designed experiments to specifically answer the following four questions:

1. How does the type of items affect COMMAND's ability to recover the prerequisite structure? Single-skilled items v.s. multi-skilled items.
2. How well does COMMAND perform when there is noise in the data? Data contains noise due to the presence of unaccounted latent variables.
3. How well does COMMAND perform when the student performance data have missing values?
4. How is COMMAND compared with other prerequisite discovery methods? Compare COMMAND to the Probabilistic Association Rules Mining (PARM) method (Chen et al., 2015).

Evaluation Metrics

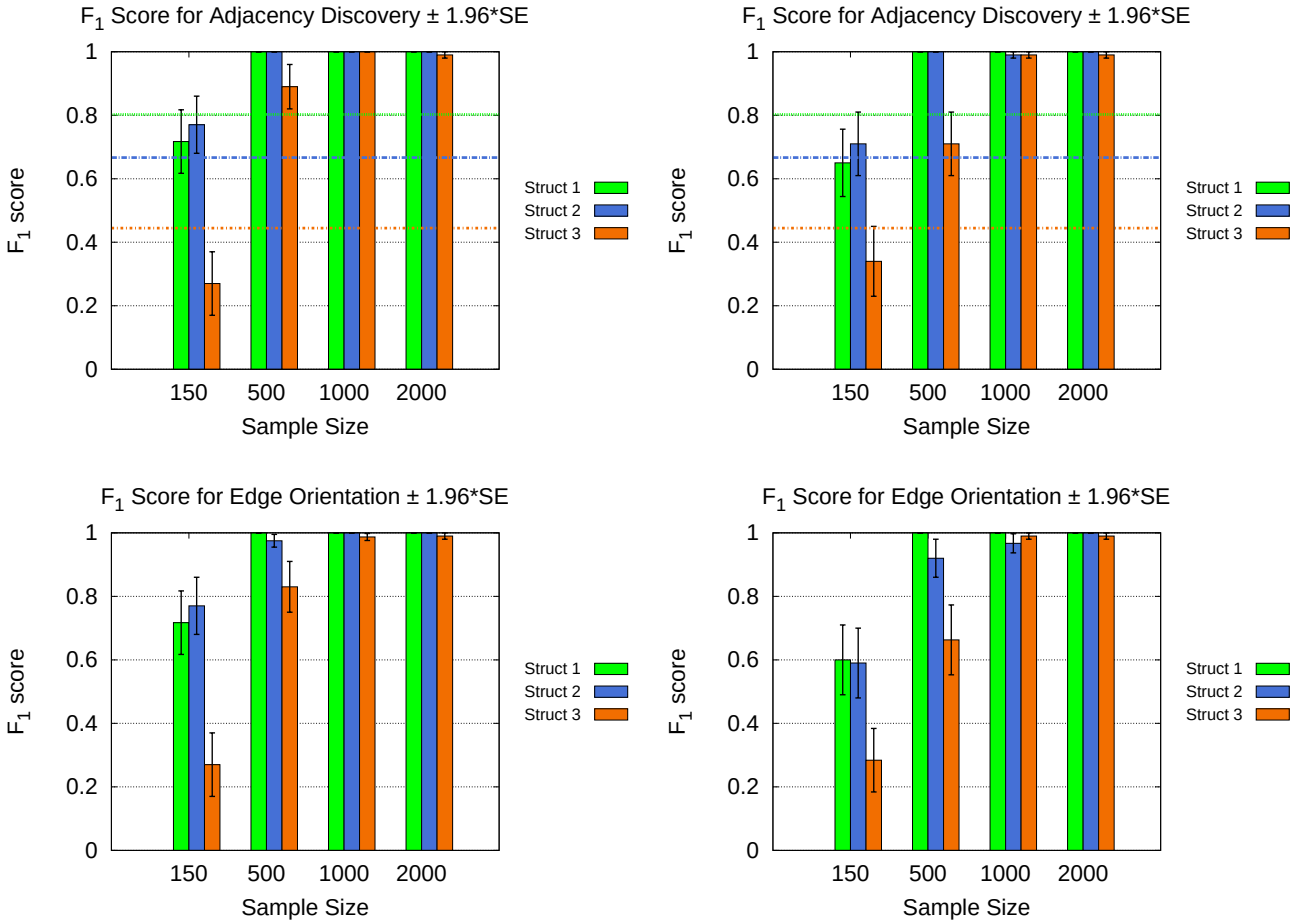
Table: Formulas for measuring adjacency rate (AR)

Metric	Formula
True positive ($TPAR$)	$\frac{\#\text{ of correct adjacencies in learned model}}{\#\text{ of adjacencies in true model}}$
True discovery ($TDAR$)	$\frac{\#\text{ of correct adjacencies in learned model}}{\#\text{ of adjacencies in learned model}}$
F_1-AR	$\frac{2 \cdot TPAR \cdot TDAR}{TPAR + TDAR}$

Table: Formulas for measuring orientation rate (OR)

Metric	Formula
True positive ($TPOR$)	$\frac{\#\text{ of correctly directed edges in learned model}}{\#\text{ of directed edges in true model}}$
True discovery ($TDOR$)	$\frac{\#\text{ of correctly directed edges in learned model}}{\#\text{ of directed edges in learned model}}$
F_1-OR	$\frac{2 \cdot TPOR \cdot TDOR}{TPOR + TDOR}$

Synthetic Data: Single-skilled vs Multi-skilled Items

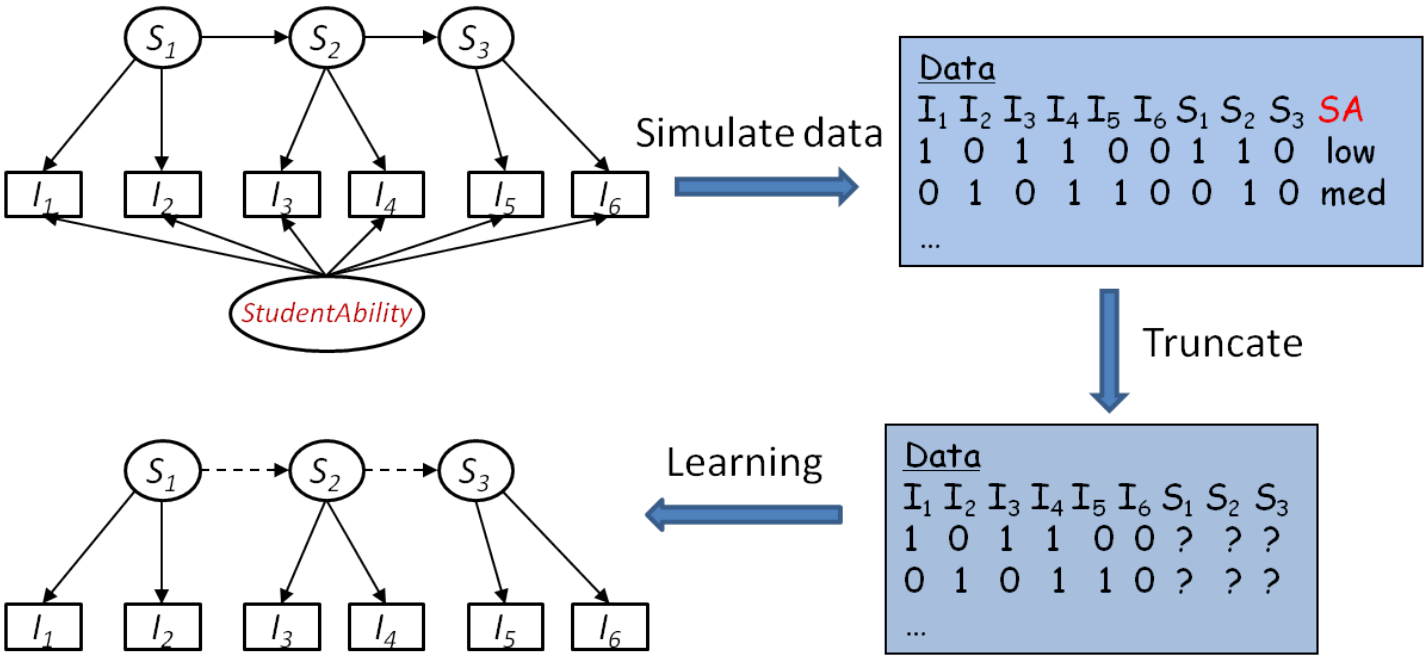


Single Skill | Multiple Skill

Figure: Comparison of F_1 scores for adjacency discovery (top row) and for edge orientation (bottom row).

Evaluation of COMMAND With Noisy Data

- Noise may occur due to the presence of latent variables that are not explicitly modeled, e.g., student ability.
- Students' performance depends not only on whether they have mastered the skills, but also on their individual ability
- Synthesized BN models including an extra variable *StudentAbility* with 3 possible states (low/med/high).



Results: Sensitivity to Noise

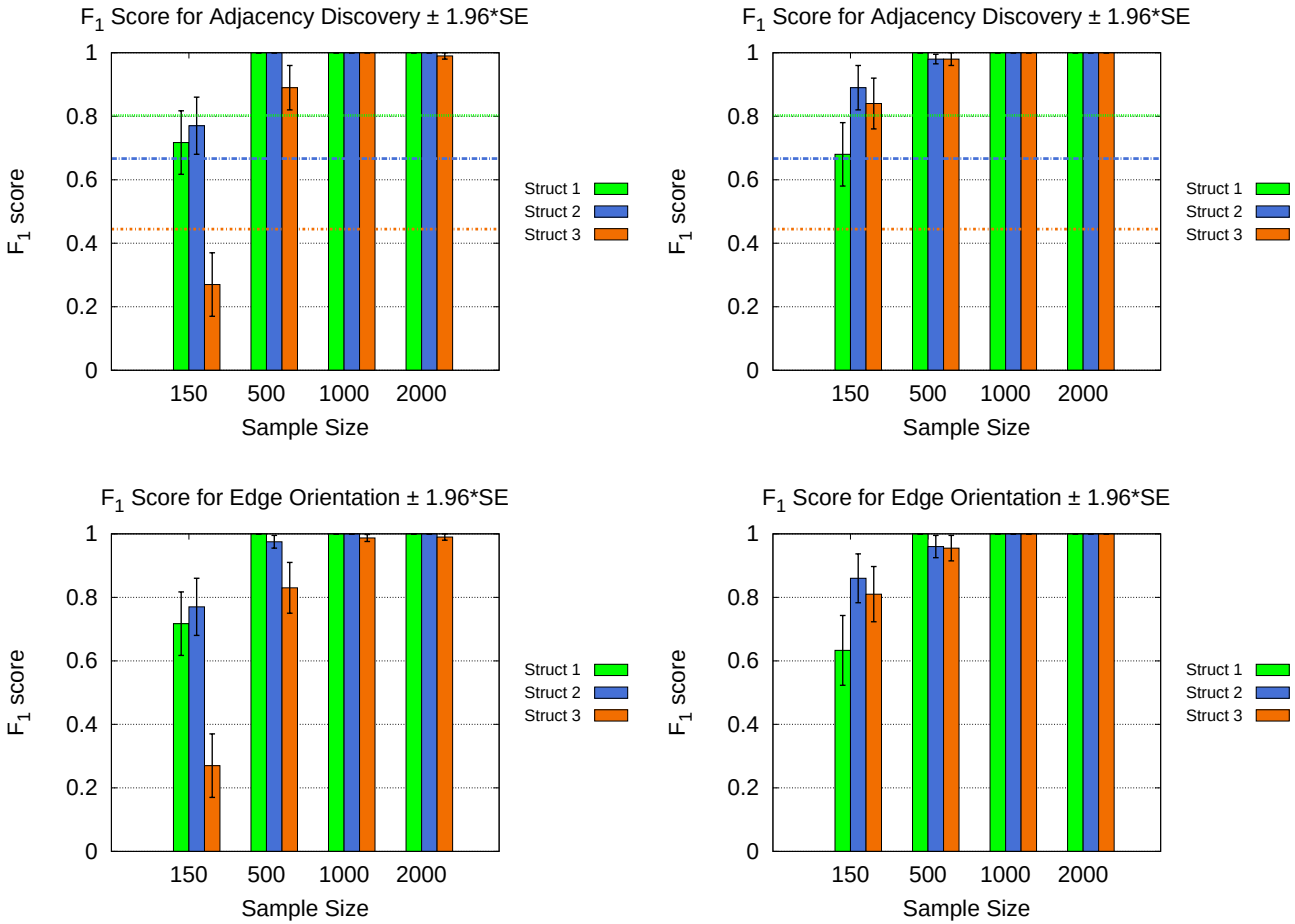


Figure: Results of adding systematic noise. Comparison of F_1 scores for adjacency discovery (top row) and for edge orientation (bottom row).

Data Containing Missing Values

- Real-world datasets collected from students often have missing values, for example, when learners do not answer all items.
- COMMAND can be applied on data containing missing values.

Table: Example student performance matrix containing missing values.

User	Item 1	Item 2	Item 3	Item p
Alice	0	?		0
Bob	?	1	...	1
Carol	0	0		?
		...		

Data Containing Missing Values

- We generated data sets of with 1000 observations with varying fraction of randomly missing values (10%, 20%, 30%, 40%, 50%).

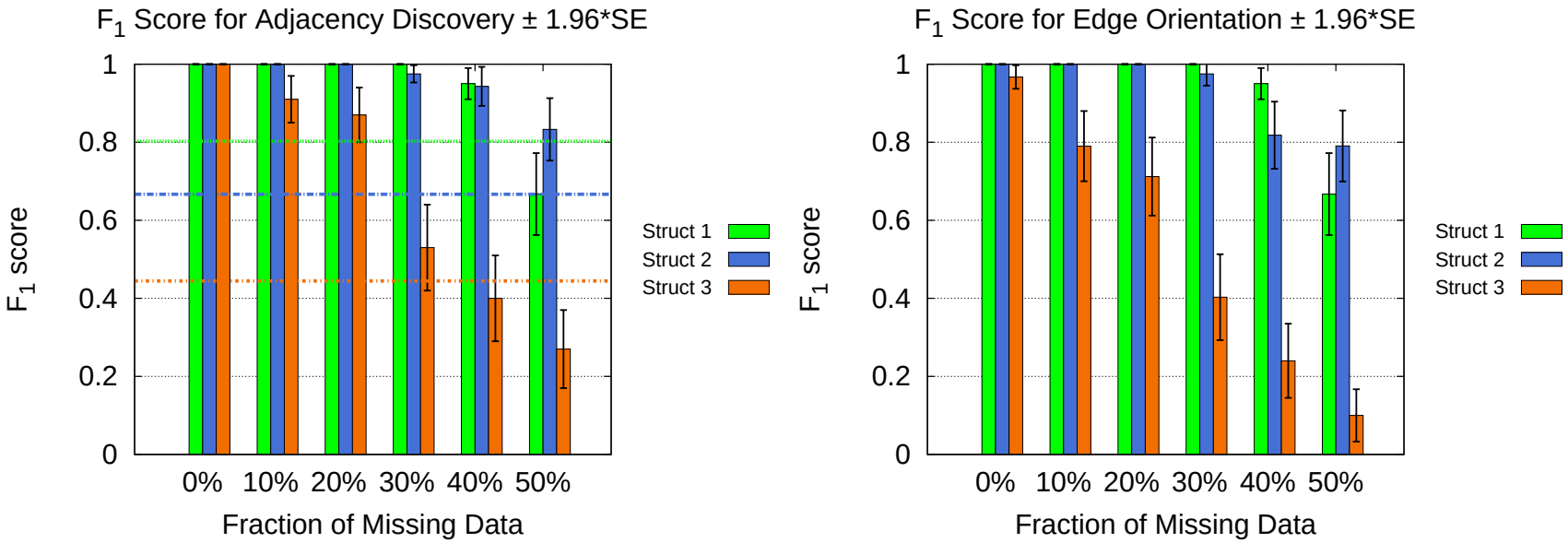
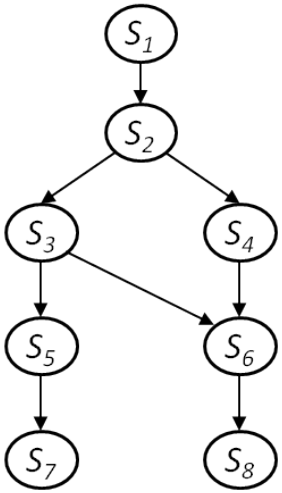


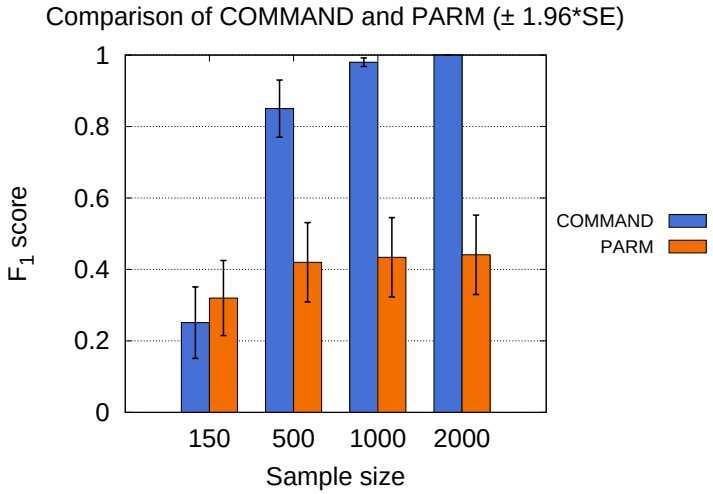
Figure: Results of learning with missing data. Comparison of F_1 scores for adjacency discovery (left) and for edge orientation (right).

Comparison With Prior Work

- Probabilistic Association Rules Mining (PARM) (Chen et al., 2015): a recent algorithm for discovering the pairwise prerequisite relationships.
- A prerequisite relationship $S_1 \rightarrow S_2$ is considered to exist if $P(S_1 = 1, S_2 = 1) \geq minsup \wedge P(S_1 = 1 | S_2 = 1) \geq minconf) \geq minprob$ and $P(P(S_1 = 0, S_2 = 0) \geq minsup \wedge P(S_2 = 0 | S_1 = 0) \geq minconf) \geq minprob$.
- Need expert to specify the thresholds *minsup*, *minconf* and *minprob*.



(a) 21 pairwise relationships



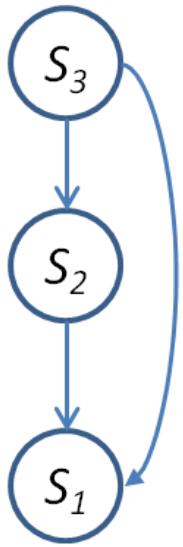
(b) *minsup* = 0.125, *minconf* = 0.76, *minprob* = 0.9 are used for PARM.

English Data Set

- The Examination for the Certification of Proficiency in English (ECPE) dataset (Templin and Bradshaw, 2014):
  - 2922 examines in their understanding of English language grammar .
  - student performance in 28 items on 3 skills (S_1: **morphosyntactic rules,** S_2: **cohesive rules, and** S_3**:lexical rules**).
  - Each item requires either one or two of the three skills.

$P(S_3=0)$	$P(S_3=1)$
0.33	0.67

S_3	$P(S_2=0 \mid S_3)$	$P(S_2=1 \mid S_3)$
0	0.90	0.10
1	0.20	0.80

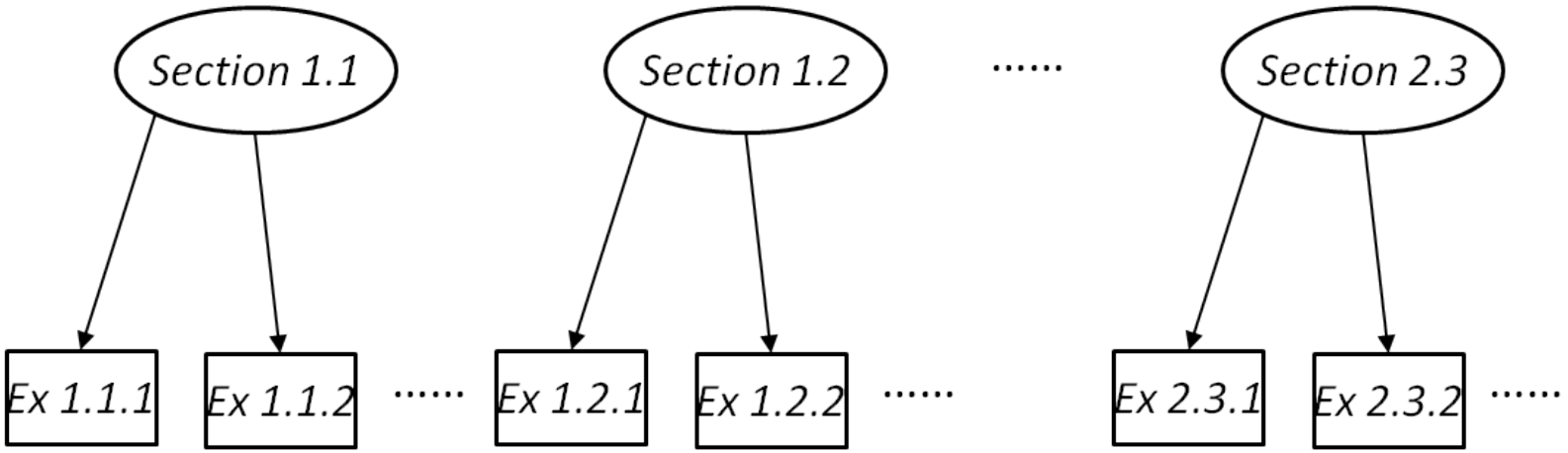


S_2	S_3	$P(S_1=0 \mid S_2,S_3)$	$P(S_1=1 \mid S_2,S_3)$
0	0	0.97	0.03
1	0	0.57	0.43
0	1	0.87	0.13
1	1	0.33	0.67

Figure: The estimated DAG and CPTs of the ECPE data set.

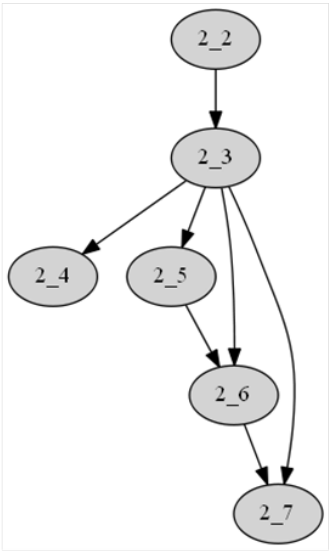
Math Data Set

- Collected from a commercial non-adaptive tutoring system.
- The textbook items are classified in chapters, sections, and objectives.
- Define skills as book sections and use a Q-matrix that assigns each exercise to a skill solely as the book section in which the item appears.



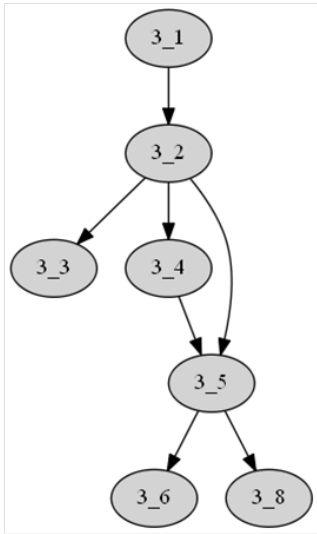
Math Data Set: Constructed Prerequisite Graph

Skill ID	Skill Name
2_2	Symbols and Sets of Numbers
2_3	Fractions and Mixed Numbers
2_4	Exponents, Order of Operations, Variable Expressions, and Equations
2_5	Adding Real Numbers
2_6	Subtracting Real Numbers
2_7	Multiplying and Dividing Real Numbers



(a) `Math-chap2`.

Skill ID	Skill Name
3_1	Simplifying Algebraic Expressions
3_2	The Addition and Multiplication Properties of Equality
3_3	Solving Linear Equations
3_4	An Introduction to Problem Solving
3_5	Formulas and Problem Solving
3_6	Percent and Mixture Problem Solving
3_8	Solving Linear Inequalities



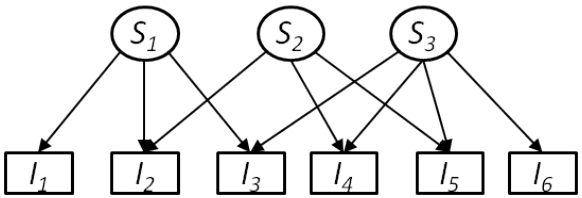
(b) `Math-chap3`.

Figure: Prerequisite structures constructed by COMMAND for `Math` data sets.

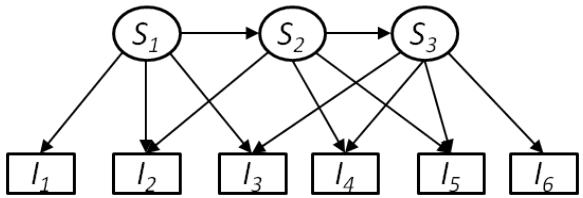
Predictive Performance

We evaluate the accuracy of the predicted student performance on an item, when we observe the student response on the other items. We compare our model with five baseline models:

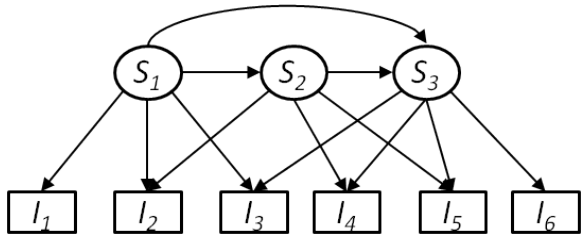
- A *majority* classifier which always classifies an instance to the majority class.
- A Bayesian network model in which the skill variables are *disconnected*.
- A Bayesian network model in which the skill variables are connected in a *chain* structure, i.e., 2-2→2-3→2-4→ . . .
- A Bayesian network model constructed using the pairwise relationships output from *PARM*.
- A *fully connected* Bayesian network where skill variables are fully connected with each other.



Disconnected

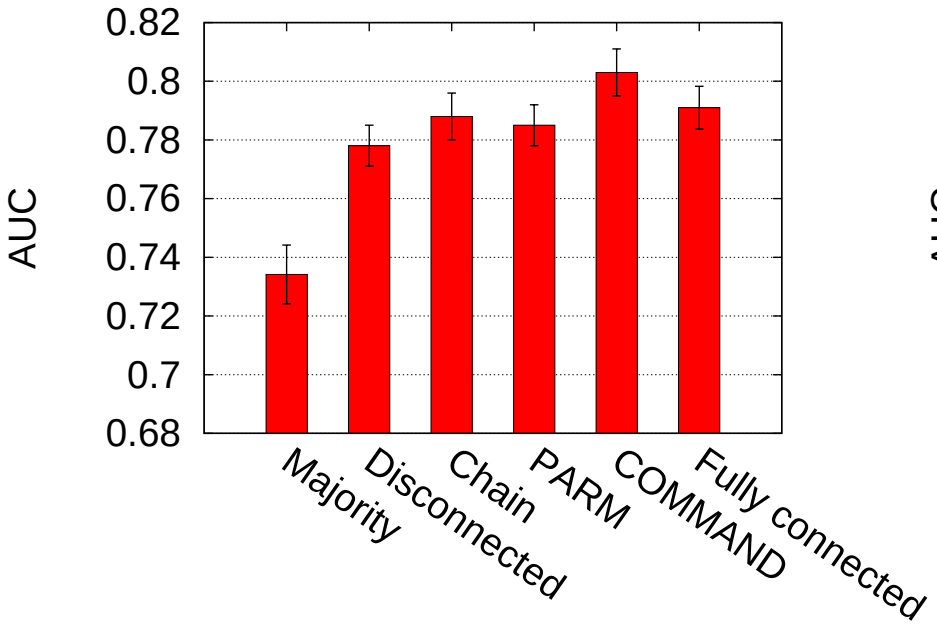


Chain

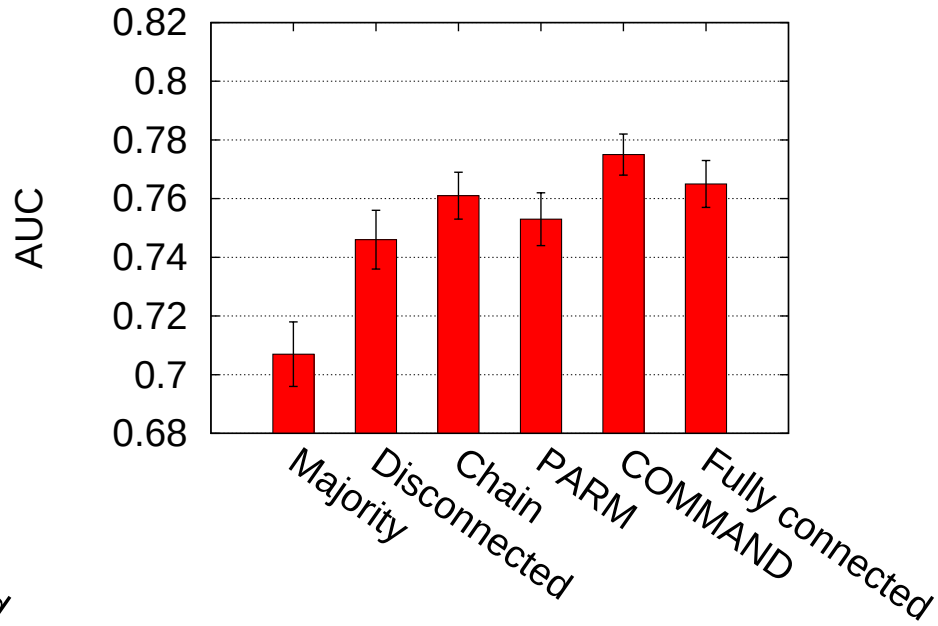


Fully connected

Predictive Performance



(a) `Math-chap2` AUC results.



(b) `Math-chap3` AUC results.

Figure: Ten fold cross-validation results of evaluating the predictions of student performance.

Conclusion

- Main contribution: a novel algorithm that simultaneously infers a prerequisite graph and a student model from data with less human intervention.
  - Optimizes the full structure of skills that captures the conditional independence between skills. Our experiments suggests that this results in better accuracy.
  - Easier to use because it does not require manual tuning of parameters.
  - Tolerates missing values in data.
- We develop a methodology to evaluate prerequisite structures on real student data.
- Learning a prerequisite graph is not merely discovering a Bayesian network— equivalent Bayesian network structures in fact represent different prerequisite structures. We proposed a theoretically motivated method to discriminate between equivalent Bayesian networks.