Curriculum Learning of Bayesian Network Structures

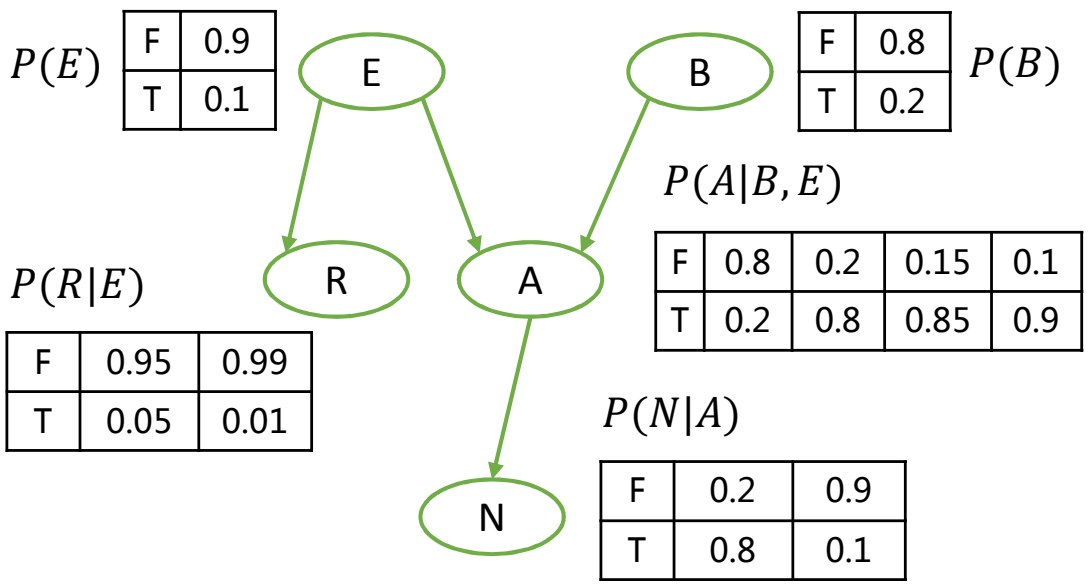
Yanpeng Zhao[1], Yetian Chen[2], Kewei Tu[3], Jin Tian[4]

{zhaoyp1[1], tukw[3]}@shanghaitech.edu.cn {yetianc[2], jtian[4]}@iastate.edu

ACML, Nov 22th, 2015, HongKong

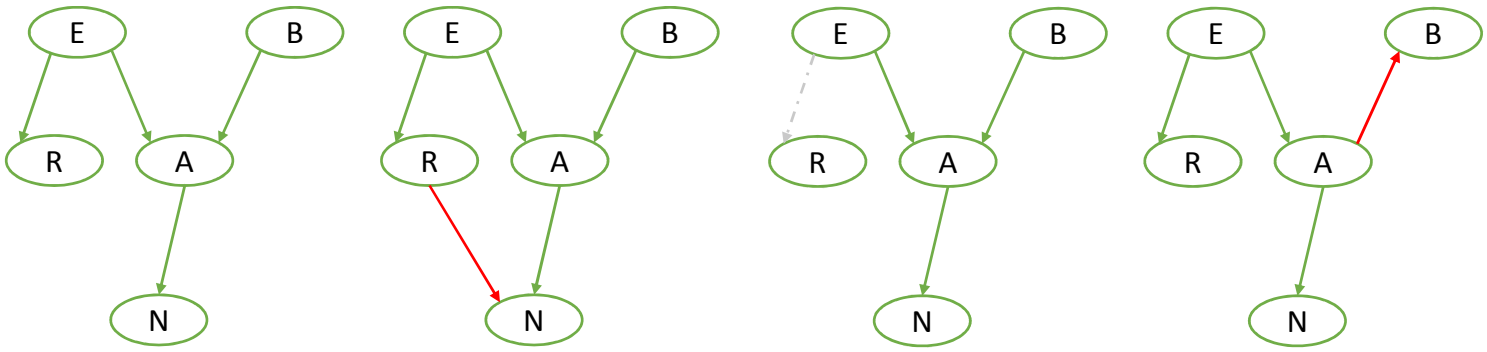
Bayesian Network

- A directed acyclic graph DAG where
  - nodes: random variables
  - directed edges: probability dependencies among variables
- An example



BN Structure Learning

- Why
  - effective inference, causal modeling
- How
  - construct the topology of the network using a structure searcher



  - score the constructed network using a scoring function

$$score(G : D) = \log P(G|D) \propto \log P(D|G) + \log P(G)$$

BN Structure Learning

$\mathcal{Q}$: How can we find the relations between all variables?

Instances

(0 0 0 1 1 1 0 0)
(0 0 0 0 0 0 1 1)
(1 0 0 1 1 1 0 0)
(0 1 0 0 1 1 1 0)
(0 0 1 1 1 0 0 0)
(1 1 0 0 0 0 1 1)
(0 1 0 1 1 1 0 0)
(0 1 0 0 1 1 0 0)
(1 0 1 1 0 1 1 0)
(0 1 0 1 1 1 0 0)

Variables

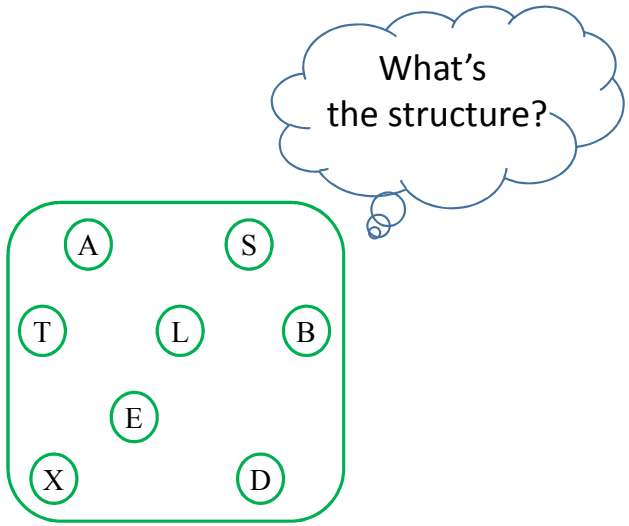
BN Structure Learning

$\mathcal{Q}$: How can we find the relations between all variables?

Instances

(0 0 0 1 1 1 0 0)
(0 0 0 0 0 0 1 1)
(1 0 0 1 1 1 0 0)
(0 1 0 0 1 1 1 0)
(0 0 1 1 1 0 0 0)
(1 1 0 0 0 0 1 1)
(0 1 0 1 1 1 0 0)
(0 1 0 0 1 1 0 0)
(1 0 1 1 0 1 1 0)
(0 1 0 1 1 1 0 0)

Variables

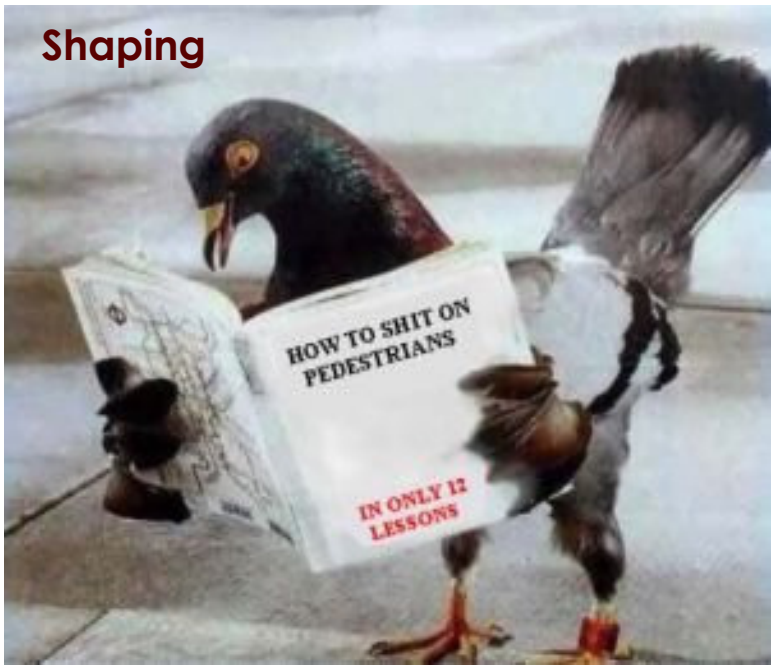


Variables S, B, D, L, E, X, A, T correspond to each column of the dataset respectively

Curriculum Learning [1]

Ideas

Guided learning helps training humans and animals



Start from simpler examples / easier tasks (Piaget 1952, Skinner 1958)

From http://www.iro.umontreal.ca/ bengioy/talks/icml2009.pdf

[1] Yoshua Bengio et al. ICML 2009

Curriculum Learning

- A curriculum is a sequence of weighting schemes of the training data $\langle W_1, W_2, \ldots, W_n \rangle$
  - W_1 assigns more weight to easier samples
  - each next scheme assigns more weight to harder samples
  - W_n assigns uniform weight to all samples
- Advantages
  - faster convergence to a minimum
  - convergence to better local minimum
- Difficulties
  - how to define better curriculum strategies

Learn BN Structure via CL

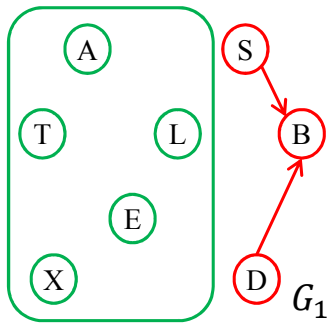
Motivation

- Human learn in a more organized way, starting with more common samples that involve dependency relations between only a small subset of variables

Instances

(0 0 0 1 1 1 0 0)
(0 0 0 0 0 0 1 1)
(1 0 0 1 1 1 0 0)
(0 1 0 0 1 1 1 0)
(0 0 1 1 1 0 0 0)
(1 1 0 0 0 0 1 1)
(0 1 0 1 1 1 0 0)
(0 1 0 0 1 1 0 0)
(1 0 1 1 0 1 1 0)
(0 1 0 1 1 1 0 0)

Variables



At stage 1, learn a subnet G_1 over $\{S, B, D\}$ from scratch with the rest variables fixed at (1 1 1 0 0)

Learn BN Structure via CL

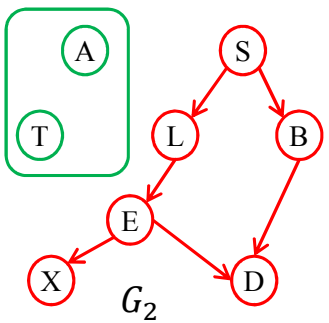
Motivation

- Only turn to less common data samples involving dependency relations with additional variables when some knowledge (i.e, a partial model) is obtained

Instances

(0 0 0 1 1 1 0 0)
(0 0 0 0 0 0 1 1)
(1 0 0 1 1 1 0 0)
(0 1 0 0 1 1 1 0)
(0 0 1 1 1 0 0 0)
(1 1 0 0 0 0 1 1)
(0 1 0 1 1 1 0 0)
(0 1 0 0 1 1 0 0)
(1 0 1 1 0 1 1 0)
(0 1 0 1 1 1 0 0)

Variables



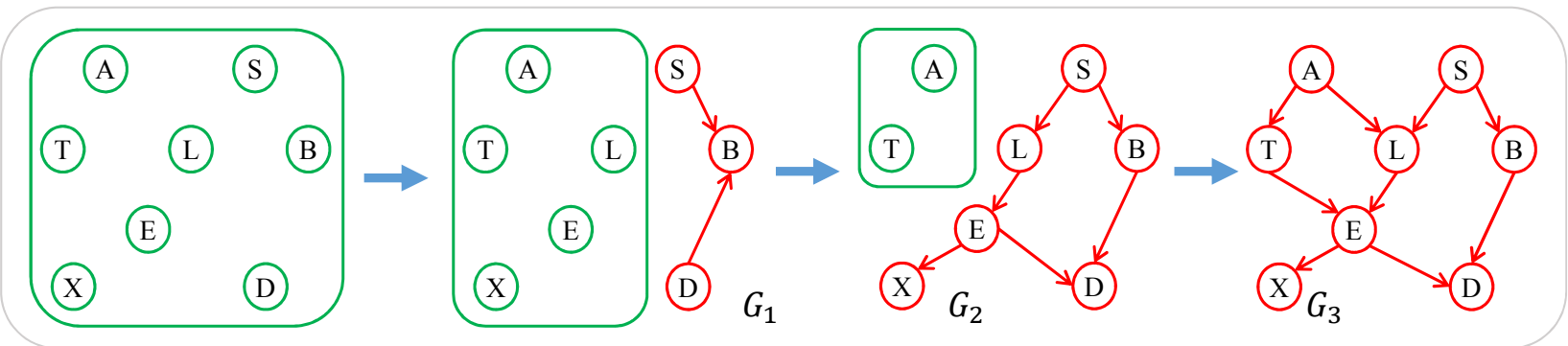
G_2

At stage 2, learn a larger subnet G_2 over $\{S, B, D, L, E, X\}$ with G_1 as the start point of search while fix the rest variables at (0 0)

Curriculum in BN Structure Learning

We define the curriculum as $(\mathbf{X}_{(1)}, ..., \mathbf{X}_{(n)})$, a sequence of selected subsets of the random variables $\mathbf{X}_{(i)}$, over which the corresponding subnet G_i is learnt.

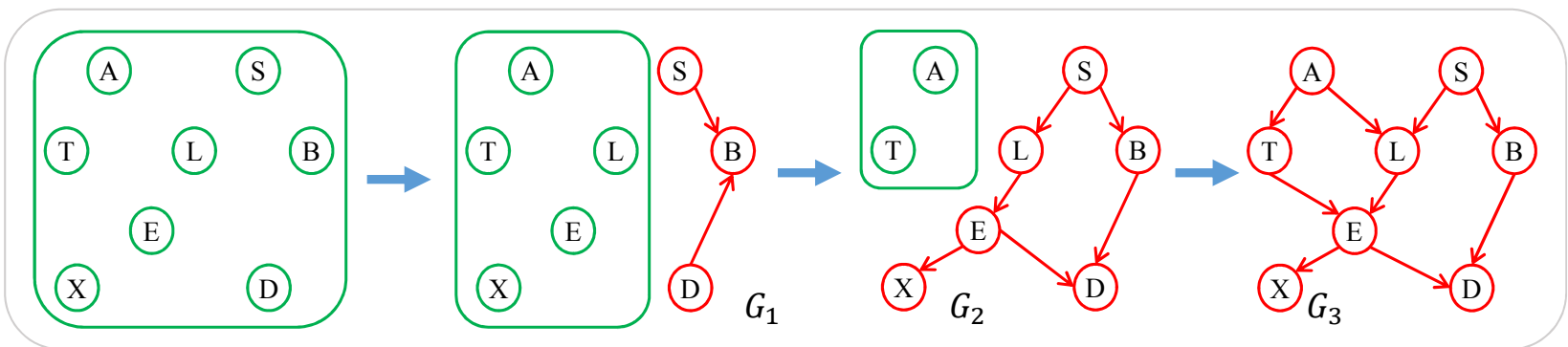
Where $\mathbf{X} = (X_1, ..., X_n)$ is a *variable set*, $\mathbf{X}_{(i)} \subseteq \mathbf{X}$, $\mathbf{X}'_{(i)} = \mathbf{X} \setminus \mathbf{X}_{(i)}$, $\mathbf{X}_{(i)} \subset \mathbf{X}_{(i+1)}$. $(G_1, ..., G_m)$ is a sequence of *intermediate learning targets.*



Curriculum in BN Structure Learning

We define the curriculum as $(\mathbf{X}_{(1)}, ..., \mathbf{X}_{(n)})$, a sequence of selected subsets of the random variables $\mathbf{X}_{(i)}$, over which the corresponding subnet G_i is learnt.

Where $\mathbf{X} = (X_1, ..., X_n)$ is a *variable set*, $\mathbf{X}_{(i)} \subseteq \mathbf{X}$, $\mathbf{X}'_{(i)} = \mathbf{X} \setminus \mathbf{X}_{(i)}$, $\mathbf{X}_{(i)} \subset \mathbf{X}_{(i+1)}$. $(G_1, ..., G_m)$ is a sequence of *intermediate learning targets.*



Curriculum: $\{\mathbf{X}_{(1)}, \mathbf{X}_{(2)}, \mathbf{X}_{(3)}\}$

$$\mathbf{X}_{(1)} = \{S, B, D\}, \mathbf{X}_{(2)} = \{S, B, D, L, E, X\}, \mathbf{X}_{(3)} = \{S, B, D, L, E, X, A, T\}\}$$

Curriculum in BN Structure Learning

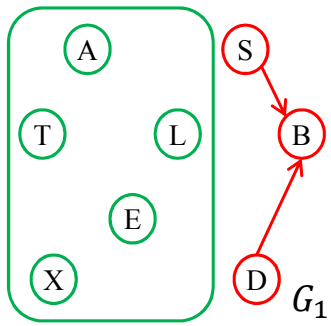
In terms of the sample weighting scheme $\langle W_1, W_2, \ldots, W_n \rangle$, at stage i, W_i assigns '1' to those samples with $\mathbf{X}'_{(i)} = \mathbf{x}'_{(i)}$ (the fixed value) and '0' to the other samples.

Instances

(0	0	0	1	1	1	0	0)		1)
(0	0	0	0	0	0	1	1)		0
(1	0	0	1	1	1	0	0)		1
(0	1	0	0	1	1	1	0)		0
(0	0	1	1	1	0	0	0)		0
(1	1	0	0	0	0	1	1)		0
(0	1	0	1	1	1	0	0)		1
(0	1	0	0	1	1	0	0)		0
(1	0	1	1	0	1	1	0)		0
(0	1	0	1	1	1	0	0)		1

Variables

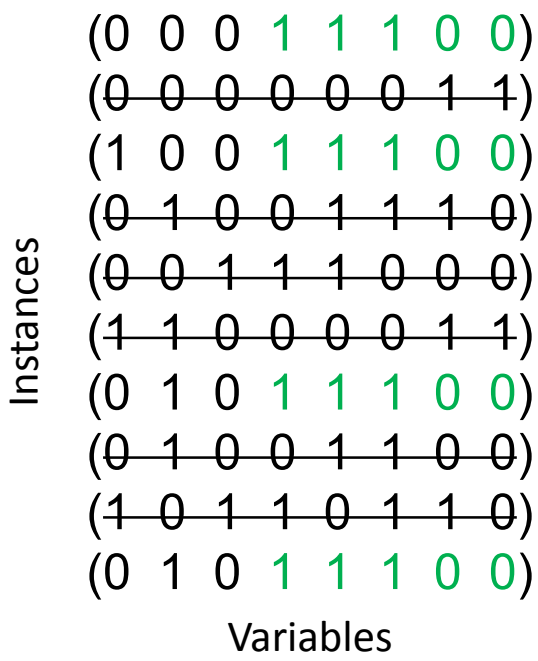
W_1

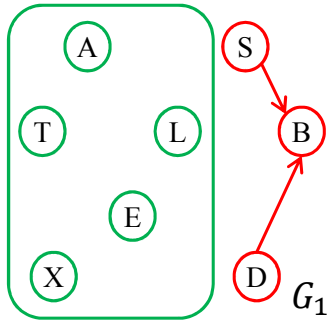


At stage 1, learn a subnet G_1 over $\mathbf{X}_{(i)} = \{S, B, D\}$ with $\mathbf{X}'_{(i)} = \{L, E, X, A, T\}$ fixed at (1 1 1 0 0). For the left dataset, W_1 = (1 0 1 0 1 0 1 0 0 1)

Limitation

- We only used a small fraction of the dataset at each learning stage

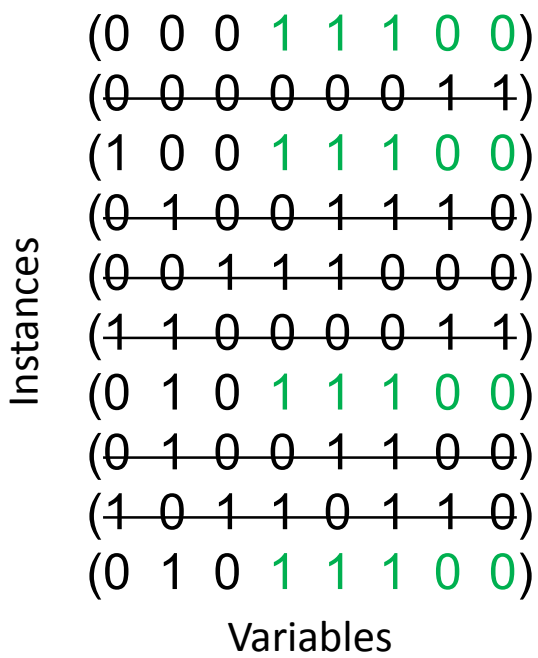


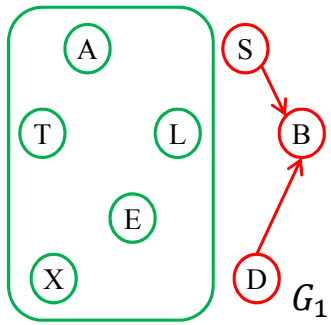


At stage 1, learn a subnet G_1 over $\mathbf{X}_{(i)} = \{S, B, D\}$. Here we only used the samples with $\mathbf{X}'_{(i)} = \{L, E, X, A, T\}$ fixed at (1 1 1 0 0), the samples with a strikeout **IS NOT** used

Limitation

- We only used a small fraction of the dataset at each learning stage





At stage 1, learn a subnet G_1 over $\mathbf{X}_{(i)} = \{S, B, D\}$. Here we only used the samples with $\mathbf{X}'_{(i)} = \{L, E, X, A, T\}$ fixed at (1 1 1 0 0), the samples with a strikeout **IS NOT** used

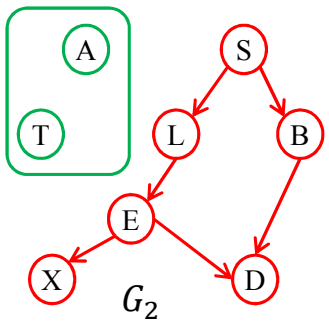
$\mathcal{Q}$: Whether we can use all the samples at a learning stage?

Solution

- An important observation
  - when we fix $\mathbf{X}'_{(i)}$ to different values, our learning target is actually the same DAG structure G_i but with different parameters (CPDs)

(0 0 0 1 1 1 0 0)
(1 0 0 1 1 1 0 0)
(0 0 1 1 1 0 0 0)
(0 1 0 1 1 1 0 0)
(0 1 0 0 1 1 0 0)
(0 1 0 1 1 1 0 0) } $D_{2,1}$
(1 1 0 0 0 0 1 1)
(0 0 0 0 0 0 1 1) } $D_{2,2}$
(1 0 1 1 0 1 1 0)
(0 1 0 0 1 1 1 0) } $D_{2,3}$

Data Segments



G_2

At stage 2, learn a subnet G_1 over $\mathbf{X}_{(2)} = \{S, B, D, L, E, X\}$. $\mathbf{X}'_{(i)}$ can take value from $\{(0,0),(1,1),(1,0)\}$

Solution

Let $\mathbf{X}'_{(i)}$ take value from $\{\mathbf{x}'_{(i),1}, ..., \mathbf{x}'_{(i),q}\}$, the set of data segments $D_i = \{D_{i,1}, ..., D_{i,q}\}$ by grouping samples based on the values of $\mathbf{X}'_{(i)}$.

- Assumption
  - $D_{i,1}, ..., D_{i,q}$ are generated by the same G_i but with *independent* CPDs

Solution

Let $\mathbf{X}'_{(i)}$ take value from $\{\mathbf{x}'_{(i),1}, ..., \mathbf{x}'_{(i),q}\}$, the set of data segments $D_i = \{D_{i,1}, ..., D_{i,q}\}$ by grouping samples based on the values of $\mathbf{X}'_{(i)}$.

- Assumption
  - $D_{i,1}, ..., D_{i,q}$ are generated by the same G_i but with *independent* CPDs
- Bayesian score function

$$\log P(G_i, D_i) = C + \sum_{j=1}^{q} \log P(G_i, D_{i,j})$$

Over-fitting

- Over-fitting occurs when sample size is small or there are many learning stages
- How to avoid

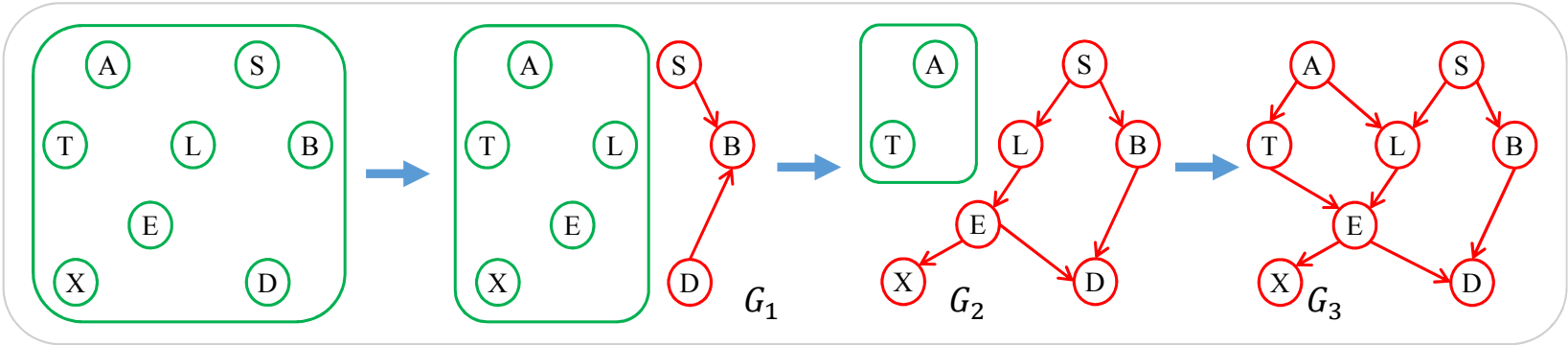
$$penalty(G_i : D_i) = \left(\frac{a}{SS} + \frac{V(G_i)}{b}\right) E(G_i),$$

SS: sample size; $V(G_i)$: number of the variables in G_i, $E(G_i)$: number of edges in G_i; a, b: positive constants.

- The final score function

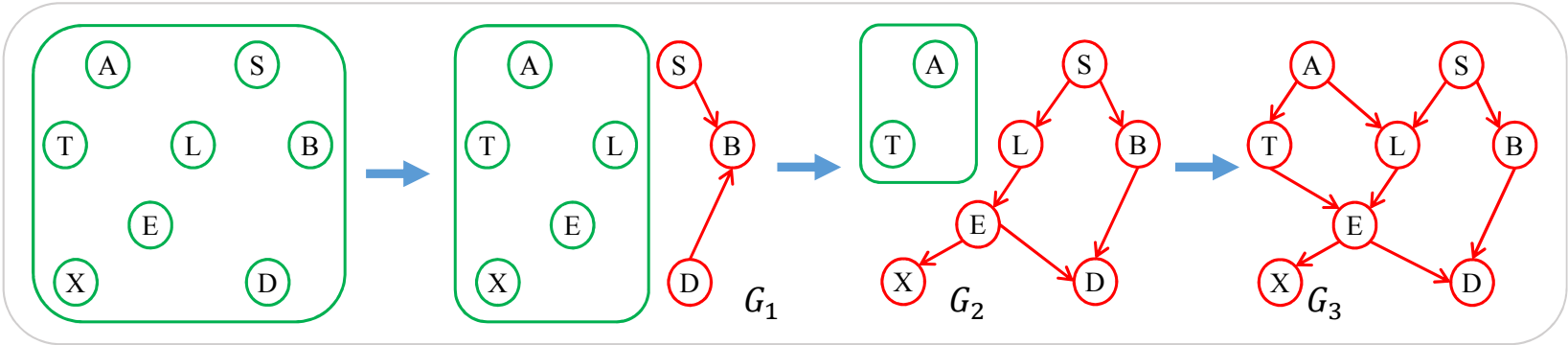
$$score(G_i : D_i) = \sum_{j=1}^{q} \log P(G_i, D_{i,j}) - penalty(G_i : D_i)$$

How to Make a Curriculum



$$\underbrace{S, B, D, L, E, X}_{\mathbf{x}_{(2)}} = \underbrace{S, B, D}_{\mathbf{x}_{(1)}} + \underbrace{L, E, X}_{\text{unfixed}}$$

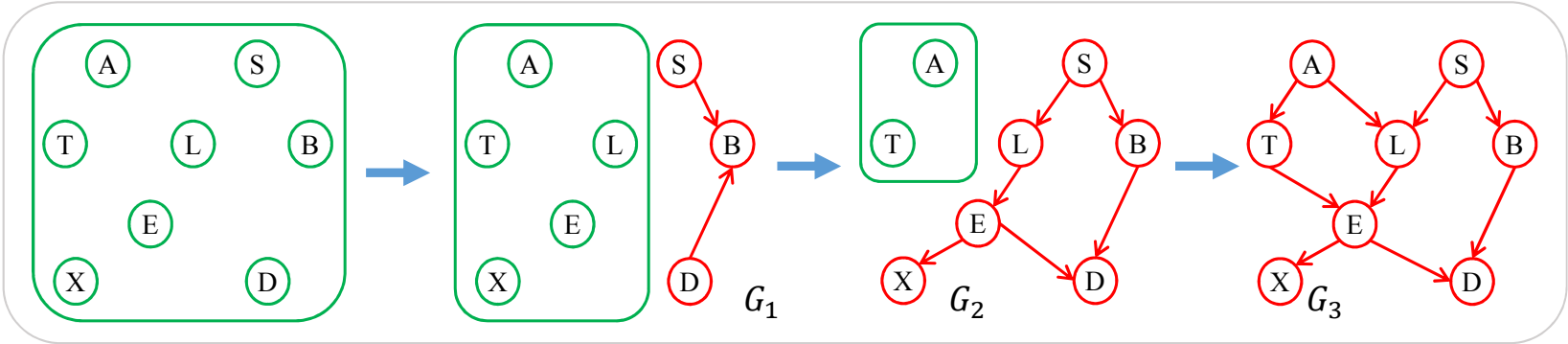
How to Make a Curriculum



$$\underbrace{S,B,D,L,E,X}_{\mathbf{X}_{(2)}} = \underbrace{S,B,D}_{\mathbf{X}_{(1)}} + \underbrace{L,E,X}_{\text{unfixed}}$$

$\mathcal{Q}$: Which variables shall be unfixed (added into $\mathbf{X}_{(i-1)}$)?

How to Make a Curriculum



$$\underbrace{S,B,D,L,E,X}_{\mathbf{X}_{(2)}} = \underbrace{S,B,D}_{\mathbf{X}_{(1)}} + \underbrace{L,E,X}_{\text{unfixed}}$$

$\mathcal{Q}$: Which variables shall be unfixed (added into $\mathbf{X}_{(i-1)}$)?

Answer: unfix the variables that are most likely to have connections with the current set of variables $\mathbf{X}_{(i-1)}$.

Theorems

Theorem 1 . For any i, j, k s.t. $1 \leq i < j < k \leq n$, we have

$$d_H(G_i, G_k) \geq d_H(G_j, G_k)$$

where $d_H(G_i, G_j)$ is the structural Hamming distance (SHD) between the structures of two BNs G_i and G_j.

Theorem 2 . For any i, j, k s.t. $1 \leq i < j < k \leq n$, we have

$$d_{TV}(G_i, G_k) \geq d_{TV}(G_j, G_k)$$

where $d_{TV}(G_i, G_j)$ is the total variation distance between the two distributions defined by the two BNs G_i and G_j.

Datasets[2]

Table : Bayesian networks used in experiments.

Network	Num. vars	Num. edges	Max in/out-degree	Cardinality range	Average cardinality
alarm	37	46	4/5	2-4	2.84
andes	223	338	6/12	2-2	2.00
asia	8	8	2/2	2-2	2.00
child	20	25	2/7	2-6	3.00
hailfinder	56	66	4/16	2-11	3.98
hepar2	70	123	6/17	2-4	2.31
insurance	27	52	3/7	2-5	3.30
sachs	11	17	3/6	3-3	3.00
water	32	66	5/3	3-4	3.63
win95pts	76	112	7/10	2-2	2.00

Cardinality denotes the number of values that a variable can take.

[2] http://www.bnlearn.com/bnrepository/

Comparison With MMHC[3]

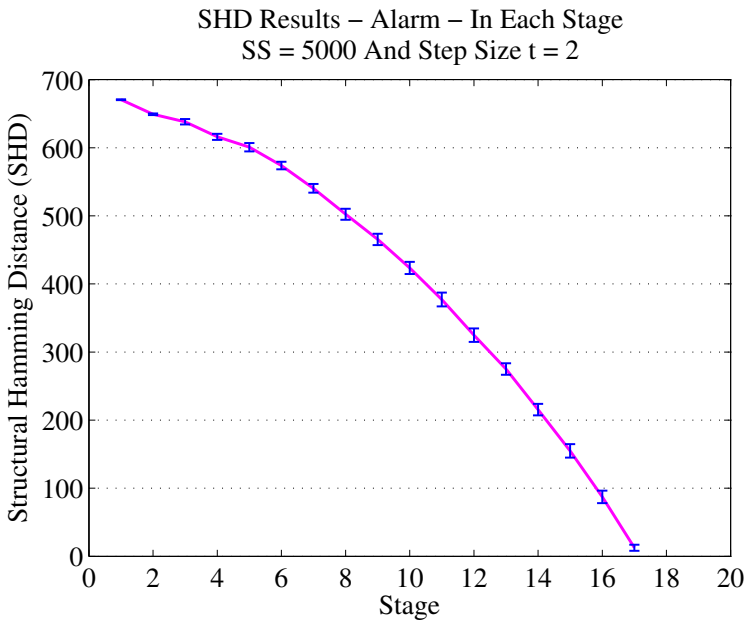
Table : Comparisons under different metrics

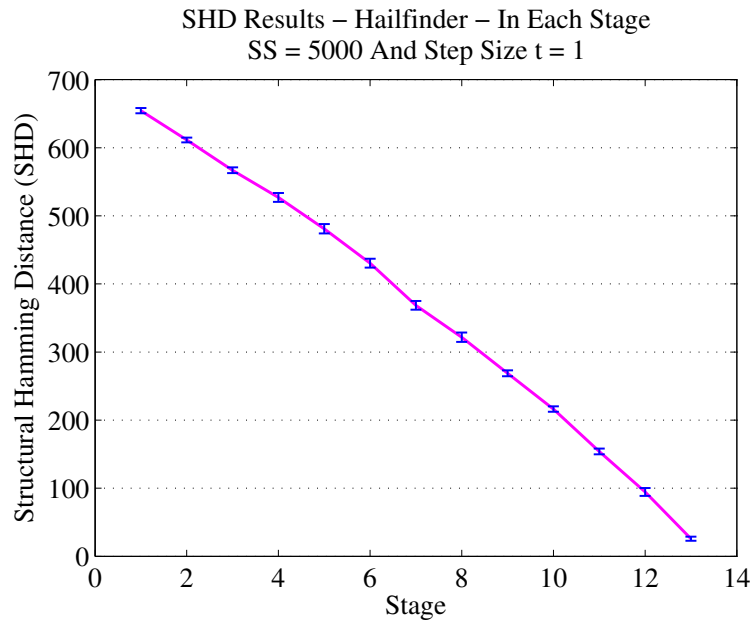
		Sample Size (SS)					
Metric	Algorithm	100	500	1000	5000	10000	50000
BDeu	CL	1(0)	**1(10)**	**1(9)**	**1(8)**	**1(10)**	**1(8)**
	MMHC	**0.89(10)**	1.06(0)	1.02(1)	1.01(2)	1.02(0)	1.01(2)
BIC	CL	1(0)	**1(9)**	**1(9)**	**1(6)**	**1(8)**	**1(8)**
	MMHC	**0.88(10)**	1.07(1)	1.02(1)	1.02(4)	1.02(2)	1.01(2)
KL	CL	1(0)	**1(10)**	**1(9)**	**1(7)**	**1(9)**	**1(9)**
	MMHC	**1.71(10)**	0.82(0)	0.96(1)	0.96(2)	0.97(0)	0.97(0)
SHD	CL	**1(7)**	**1(9)**	**1(7)**	**1(7)**	**1(8)**	**1(6)**
	MMHC	1.06(3)	1.26(1)	1.29(3)	1.07(2)	1.21(1)	1.24(3)

Results are collected under the metrics of BDeu, BIC, KL and SHD
the bold numbers represent better performance.

[3] Ioannis Tsamardinos et al. ML 2006

Verification of Theorem 1





Changes of SHD from the target BN during curriculum learning with $SS = 5000$ on the alarm and hailfinder networks.

Conclusion

- We proposed a curriculum learning algorithm for BN structure learning
- We tailored the bayesian scoring function for our algorithm
- We proved two theorems that show theoretical properties of our algorithm
- We empirically showed that our algorithm outperformed the state-of-the-art MMHC algorithm in learning BN structures

Thanks

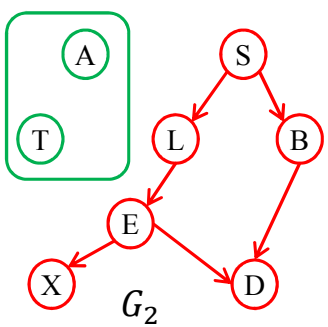
Make a Curriculum

$\mathcal{Q}$-1.How to estimate the likeliness of connections?

- By measuring the strength of the dependency (e.g., using mutual information) with the current set of variables

At stage i, compute pairwise mutual information $MI(X, Y)$ between any node X in $\mathbf{X}_{(i-1)}$ and node Y in $\mathbf{X} \setminus \mathbf{X}_{(i-1)}$. Then for any node Y in $\mathbf{X} \setminus \mathbf{X}_{(i-1)}$, compute the average pairwise mutual information by

$$AveMI(Y, \mathbf{X}_{(i-1)}) = \sum_{X \in \mathbf{X}_{(i-1)}} MI(X, Y)/|\mathbf{X}_{(i-1)}|$$

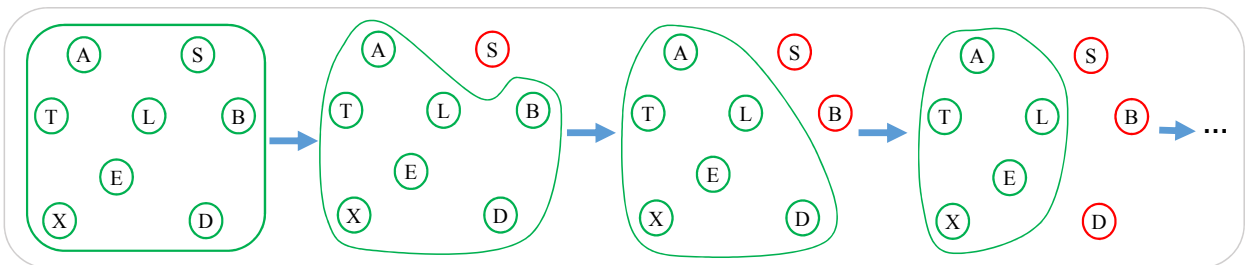


As to variable A, compute MI between A and each variable in $\{S, B, D, L, E, X\}$, then average MIs

Make a Curriculum

$\mathcal{Q}$-2.How to configure the curriculum?

1. first pick variable Y_1 with the largest $AveMI(Y_1, \mathbf{X}_{(i-1)})$
2. then pick the second variable Y_2 with the largest $AveMI(Y_2, \mathbf{X}_{(i-1)} \cup \{Y_1\})$
3. so on and so forth



First compute every variable's *AveMI* with all of the rest ones, then pick the variable *S* with largest *AveMI* and add it into a list *L*. The rest are selected in the sequential way as described above. *L* changes as this

$$(S) \rightarrow (S, B) \rightarrow (S, B, D) \rightarrow \cdots$$

Make a Curriculum

$\mathcal{Q}$-3.How many variables would be added at a stage?

- That is, from stage $i-1$ to i, which variables $\mathbf{X}_{(i-1,i)}$ should we select to produce $\mathbf{X}_{(i)} = \mathbf{X}_{(i-1)} \cup \mathbf{X}_{(i-1,i)}$?
- The number of variables selected, $|\mathbf{X}_{(i-1,i)}|$, is called the *step size*

Recall that we have got $L = (S, B, D, L, E, X, A, T)$, and every variable is most likely to have connections with the variables before it. Given *step size* 2, our curriculum could be $(\{S, B\}, \{S, B, D, L\}, \{S, B, D, L, E, X\}, \{S, B, D, L, E, X, A, T\})$

- Intuitively, the smaller step size, the more cautious and less time-efficient the algorithm is

Algorithm

Algorithm 1: Curriculum Learning of BN Structures

Input: Variable Set $\mathbf{X}$, Training Data D, Curriculum $(\mathbf{X}_{(1)}, ..., \mathbf{X}_{(m)})$.
G_0 is initialized to a network containing variables in $\mathbf{X}_{(1)}$ with no edge.
for $i = 1 \ldots m$ **do**
    Generate the set of data segments $D_i = \{D_{i,1}, ..., D_{i,q}\}$ based on the values of $\mathbf{X} \setminus \mathbf{X}_{(i)}$
    $G_i \leftarrow search(D_i, \mathbf{X}_{(i)}, G_{i-1})$
end
Return: G_m.