Causal Discovery of Dynamic Bayesian Networks

Cora Perez-Ariza[1], Ann Nicholson[2],
Kevin Korb[2], Steven Mascaro[2] and Chao Heng Hu[2]

[1] Dept. of Computer Science and Artificial Intelligence
University of Granada

[2] Clayton School of IT
Monash University

Learning Static BNs

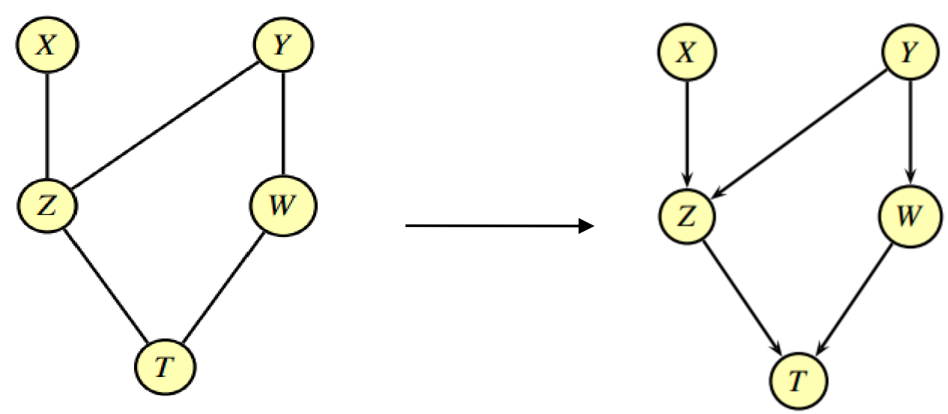
Constraint-based learning

- Performs independence tests
- e.g. PC algorithm (Spirtes *et al.,* 1993)
- Tests all pairs for direct dependencies

Learning Static BNs

Constraint-based learning

- Performs independence tests
- e.g. PC algorithm (Spirtes *et al.,* 1993)
- Tests all pairs for direct dependencies



Finding a graph pattern

Finding head to head arcs and orient the rest

Learning Static BNs

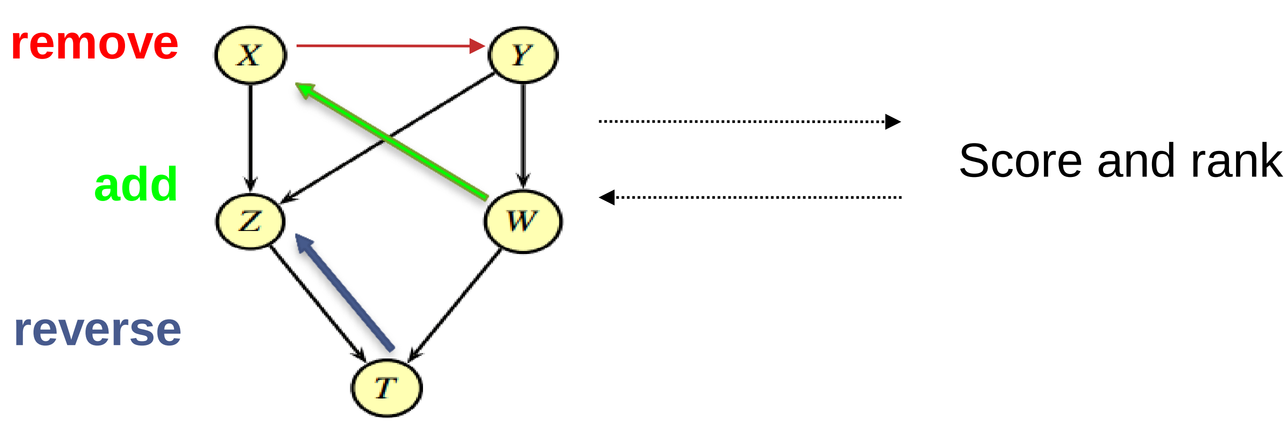
Metric-based learning

- (Stochastic) search and score
- Learning programs/packages: e.g. CaMML (Causal discovery via MML), BNT (Bayes Net Toolbox).

Learning Static BNs

Metric-based learning

- (Stochastic) search and score
- Learning programs/packages: e.g. CaMML (Causal discovery via MML), BNT (Bayes Net Toolbox).



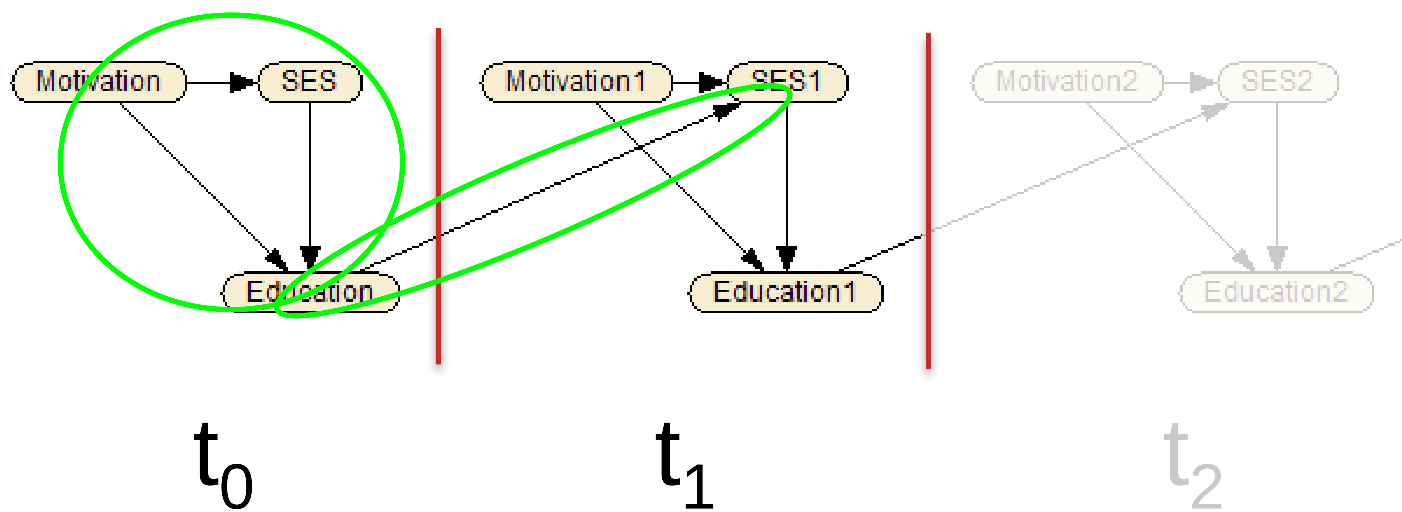
M → M' : Add/remove/reverse arcs

Dynamic Bayesian Networks

Extension of BN with arcs from t → t + 1

DBNs that we consider:

1. Same structure for each slice (i.e. stationary)
2. Arcs cannot span more than one time step



Learning *Dynamic* Bayesian Networks

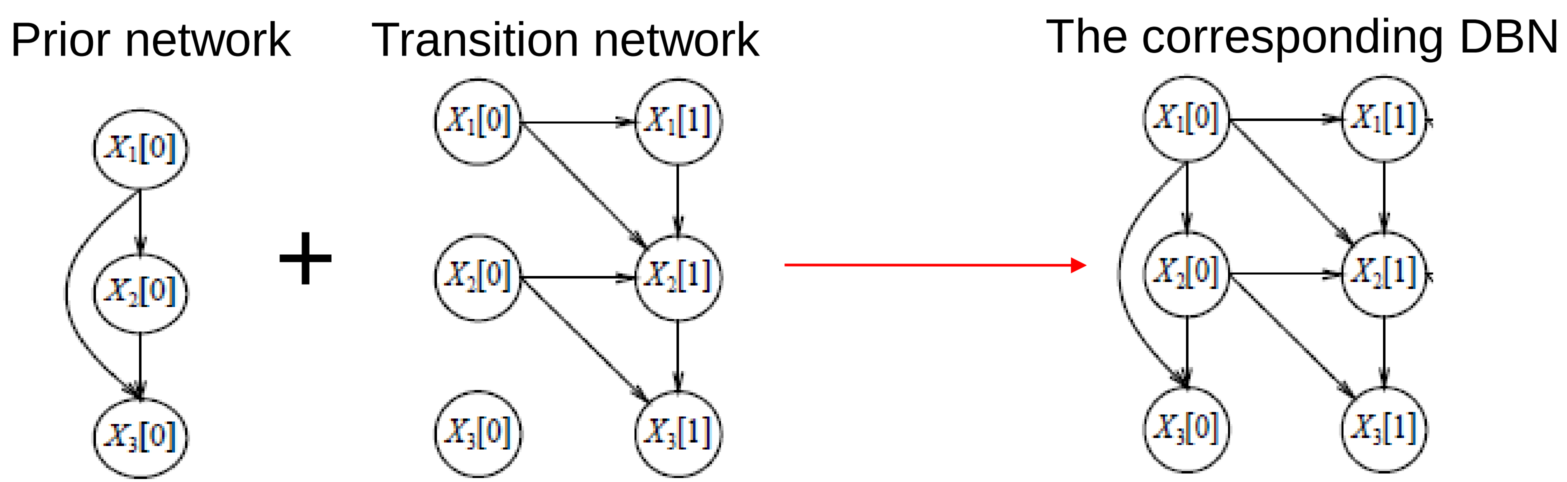
Why not use existing static learners?

- Need to guarantee slice *t* nodes come before slice *t*+1 nodes
- Often want slices to be the same (i.e. stationary)
- **Make the search more efficient**
  - ⇨ **Produce better models**

Learning DBNs – Previous Approaches

Friedman *et al.* (1998)

- Uses BIC/BDe scoring
- Hill-climbing
- Learn the prior/initial network and the transition network



Learning DBNs – Previous Approaches

Bayes Net Toolbox (BNT)

- Written by Kevin Murphy (2001)
- Supports DBN learning and inference

BNT algorithm

- Uses BIC/ML scoring
- Guarantees that $\mathbf{X}^t \prec \mathbf{X}^{t+1}$
- Only learns arcs between slices (temporal arcs)

Two New Approaches to Learning DBNs

1. Enforce stationary DBN structure with structural priors
2. Enhance existing search and score procedure to take DBN structure into account

Both take advantage of our BN learner software CaMML

CaMML

- Bayesian network learner created at Monash
- Uses MML for score and MCMC for search
- Can specify flexible priors:
  - A -> B: Direct causal connection
  - A – B: Direct relation
  - A => B: Ancestral relation
  - A ~ B: Correlation
  - Tiers
  - Existing BN structure

CaMML

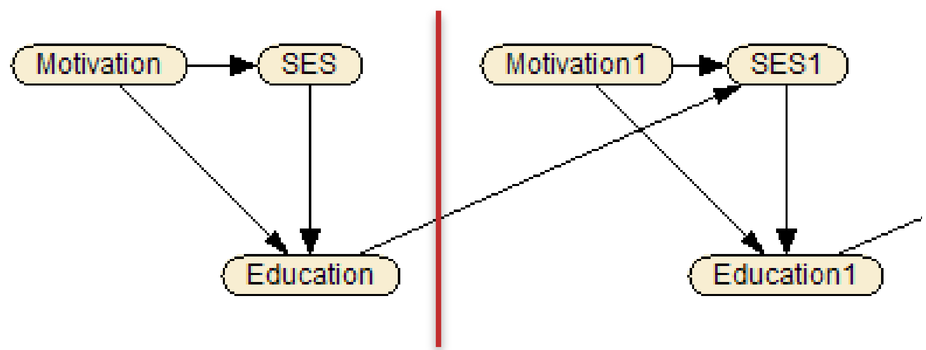
- Bayesian network learner created at Monash
- Uses MML for score and MCMC for search
- Can specify flexible priors:
  - A -> B: Direct causal connection
  - A – B: Direct relation
  - A => B: Ancestral relation
  - A ~ B: Correlation
  - **Tiers**
  - **Existing BN structure**

CaMML Tier Priors Learning

	Dataset			
	X1	X2	...	Xn
1				
2				
3				
4				
...				
...				
...				
...				

	Timestep 1					Timestep 2			
	X1	X2	...	Xn		X1	X2	...	Xn
1					2				
2					3				
3					4				
4					...				
...					...				
...					...				
...					...				
...					...				

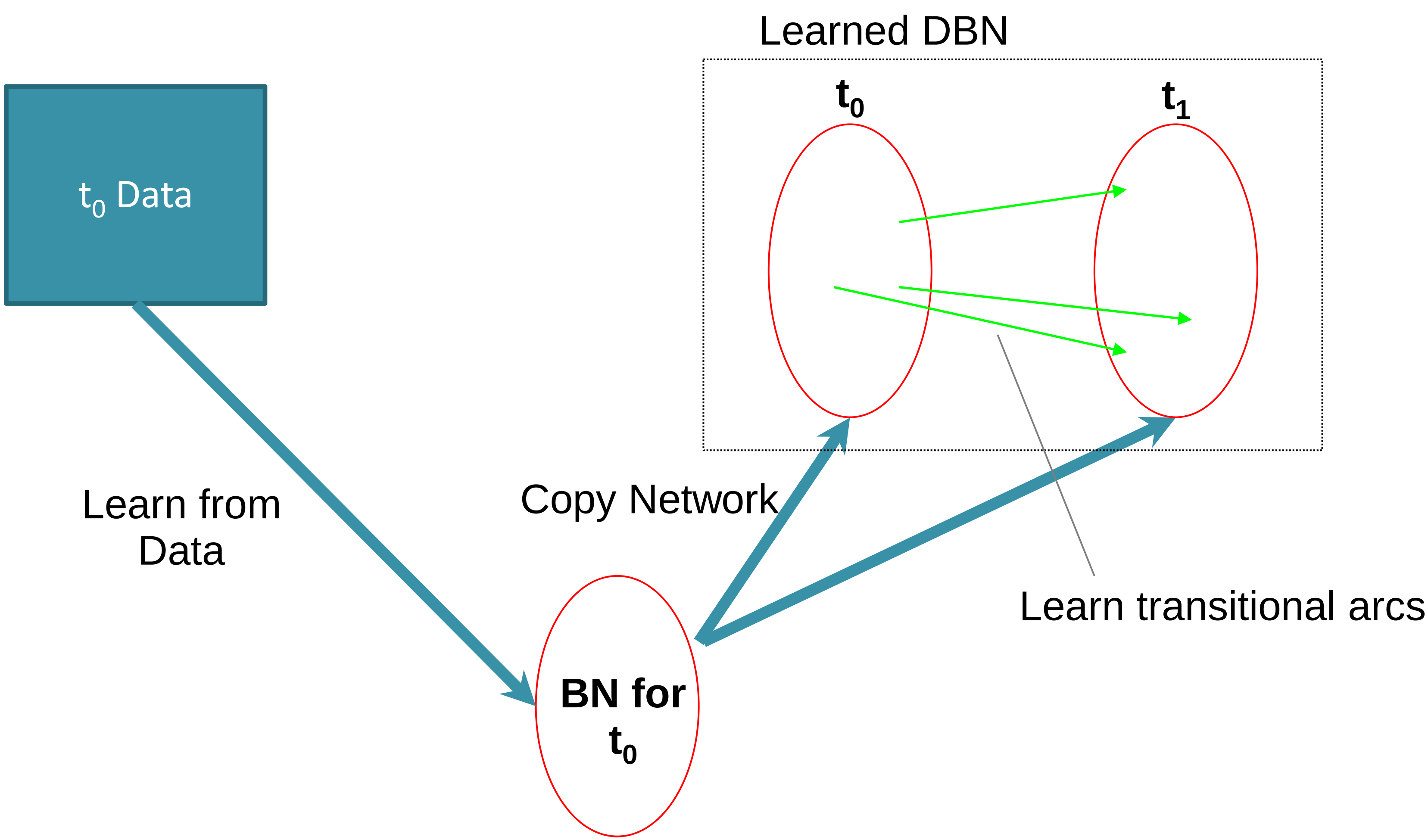
CaMML Tier Priors Learning



$t_0 \prec t_1$

Motivation, SES, Education $\prec$ **Motivation1, SES1, Education1**

CaMML 2-Step Learning



Experiments

Test models

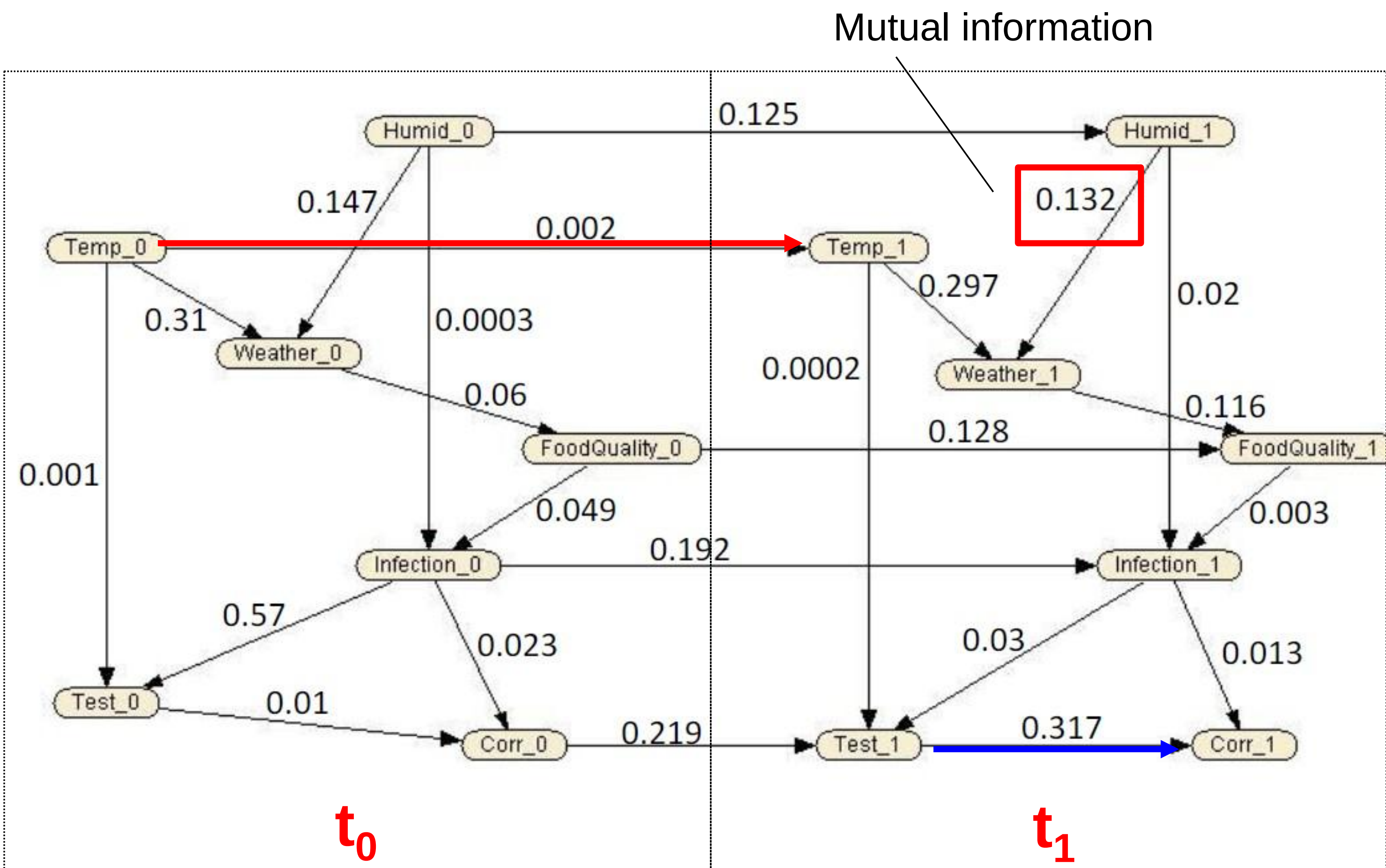
Models	Domain	# Nodes	# Arcs
Milk Infection	Agricultural	14	23
Metastatic Cancer	Health	12	18
BAT	Transport	56	68

We compare CaMML against two other learning programs:

- PC algorithm (in GeNIe)

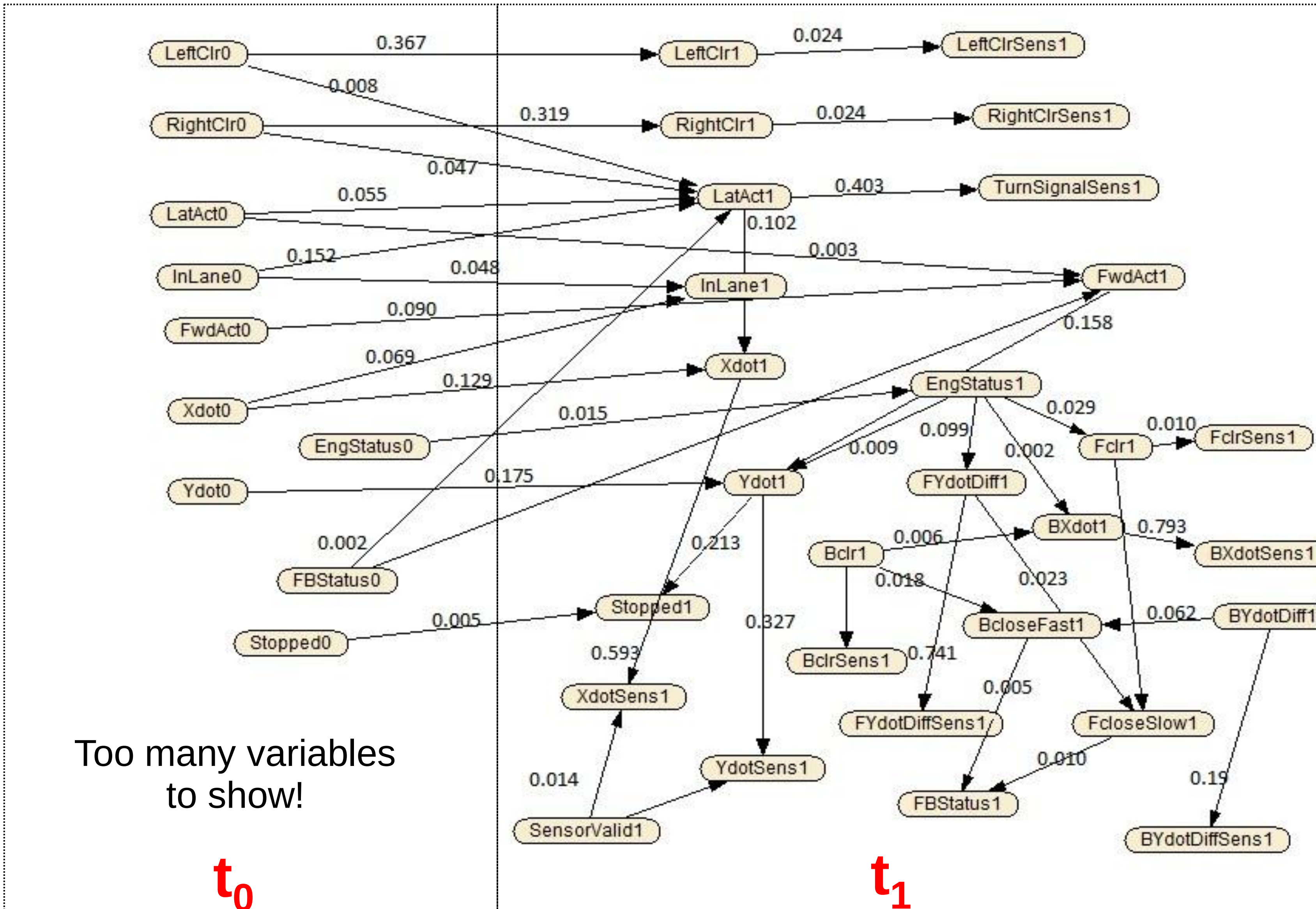
- BNT

Milk Infection DBN



Use mutual information to score strength of arcs

BAT DBN


LeftClr0
RightClr0
LatAct0
InLane0
FwdAct0
Xdot0
EngStatus0
Ydot0
FBStatus0
Stopped0
LeftClr1
LeftClrSens1
RightClr1
RightClrSens1
LatAct1
TurnSignalSens1
InLane1
FwdAct1
Xdot1
EngStatus1
Ydot1
FYdotDiff1
Fclr1
FclrSens1
BXdot1
BXdotSens1
Bclr1
BclrSens1
BcloseFast1
BYdotDiff1
Stopped1
XdotSens1
YdotSens1
FYdotDiffSens1
FcloseSlow1
FBStatus1
BYdotDiffSens1
SensorValid1
0.367
0.008
0.319
0.047
0.055
0.152
0.048
0.090
0.069
0.129
0.015
0.175
0.002
0.005
0.024
0.024
0.403
0.102
0.003
0.158
0.029
0.010
0.099
0.009
0.002
0.006
0.793
0.213
0.018
0.023
0.062
0.327
0.593
0.741
0.005
0.010
0.014
0.19
Too many variables to show!
t_0
t_1

Experiment #1

- Plain static BN learning (without using priors)
- CaMML vs GeNIe (PC algorithm)

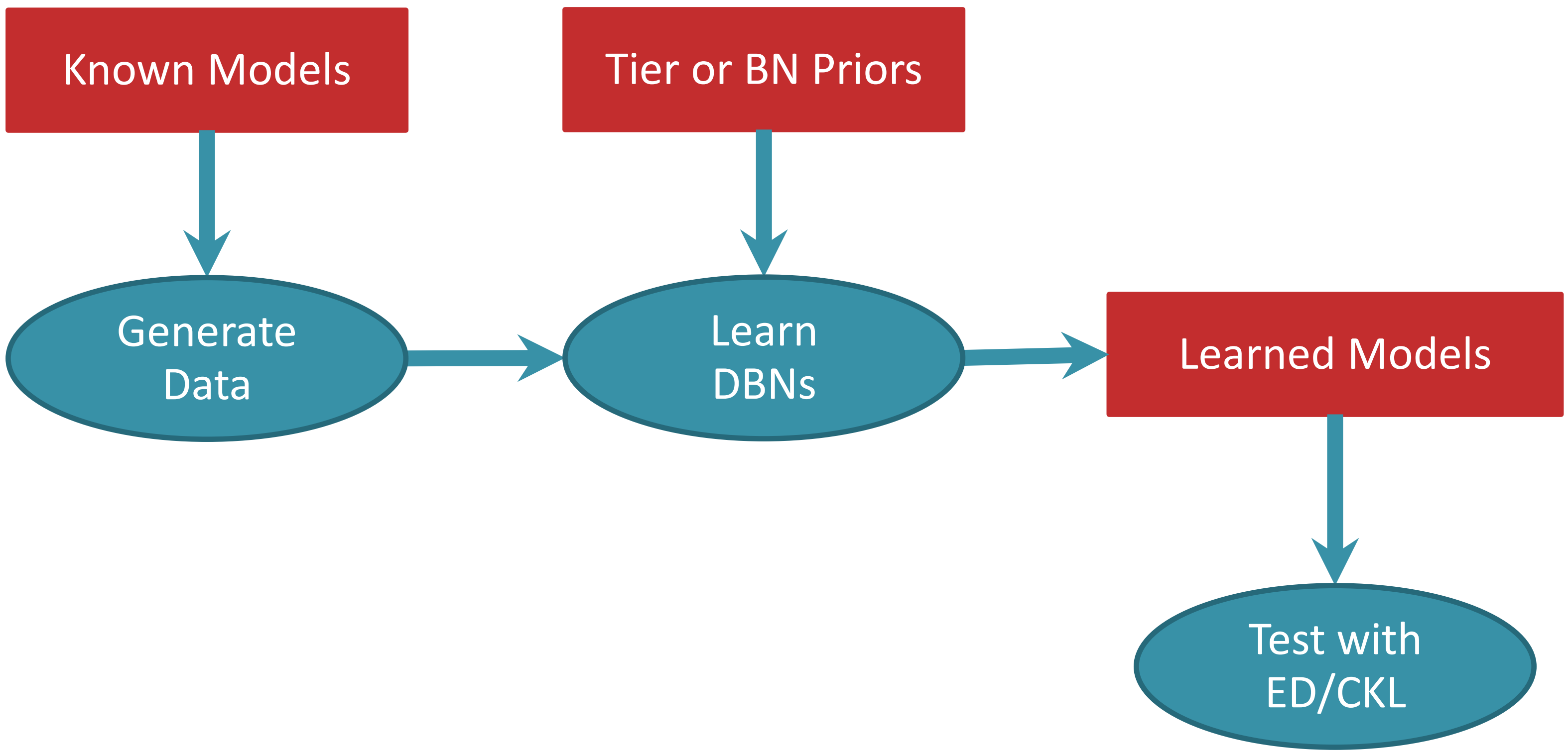
Experiment #2

- Learning with tier priors
- CaMML vs GeNIe and BNT

Experiment #3

- Learning using tier priors vs 2-step algorithm

Experiment Procedure



Evaluation

Edit distance

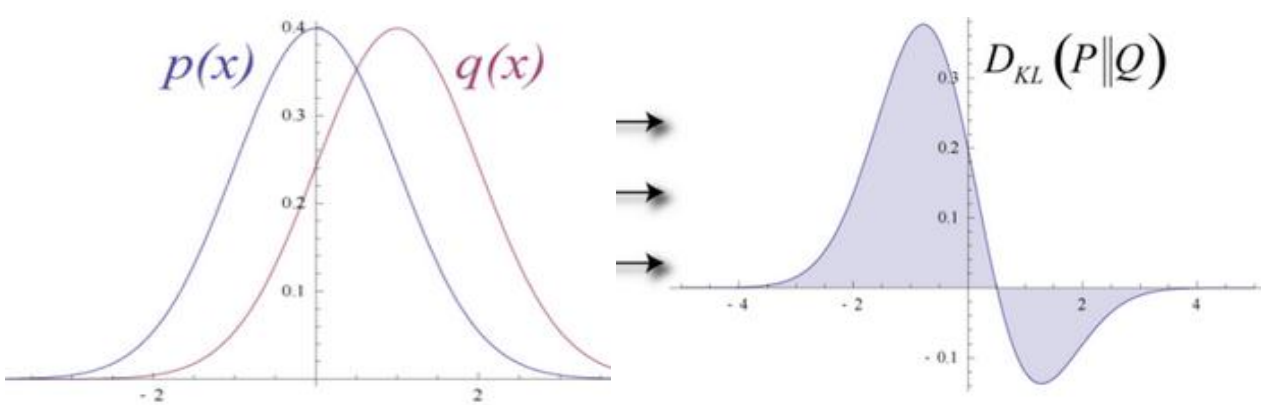
Count 1 if an arc is missing/added/reversed in the learned model

Our modification for DBNs:

$$ED_{DBN} = W_s.N_s + W_t.N_t$$

Causal Kullback-Leibler divergence

Computes the distance of probability distribution between model P and model Q



Results

Milk and cancer model (CaMML vs GeNIe, using tiers)

CaMML		temporal link errors	static link (T1) errors	total link errors
Models	Datasize			
Metastatic Cancer	100	1.3 (0.458)	4.0 (0.0)	5.3(0.458)
	1000	0.0 (0.0)	1.7 (0.458)	1.7 (0.458)
	10000	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
Milk Infection	100	4.6 (0.489)	5.9 (0.3)	10.5 (0.5)
	1000	2.0 (0.0)	1.1 (0.7)	3.1 (0.7)
	10000	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

GeNIe		temporal link errors	static link (T1) errors	total link errors
Models	Datasize			
Metastatic Cancer	100	5	4	9
	1000	3	2	5
	10000	0	2.5	2.5
Milk Infection	100	2	7	9
	1000	1	6	7
	10000	1	5	6

Results

Milk and cancer model (CaMML vs GeNIe, using tiers)

CaMML		temporal	static	total
Models	Datasize	link errors	link (T1) errors	link errors
Metastatic Cancer	100	1.3 (0.458)	4.0 (0.0)	5.3(0.458)
	1000	0.0 (0.0)	1.7 (0.458)	1.7 (0.458)
	10000	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
Milk Infection	100	4.6 (0.489)	5.9 (0.3)	10.5 (0.5)
	1000	2.0 (0.0)	1.1 (0.7)	3.1 (0.7)
	10000	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

GeNIe		temporal	static	total
Models	Datasize	link errors	link (T1) errors	link errors
Metastatic Cancer	100	5	4	9
	1000	3	2	5
	10000	0	2.5	2.5
Milk Infection	100	2	7	9
	1000	1	6	7
	10000	1	5	6

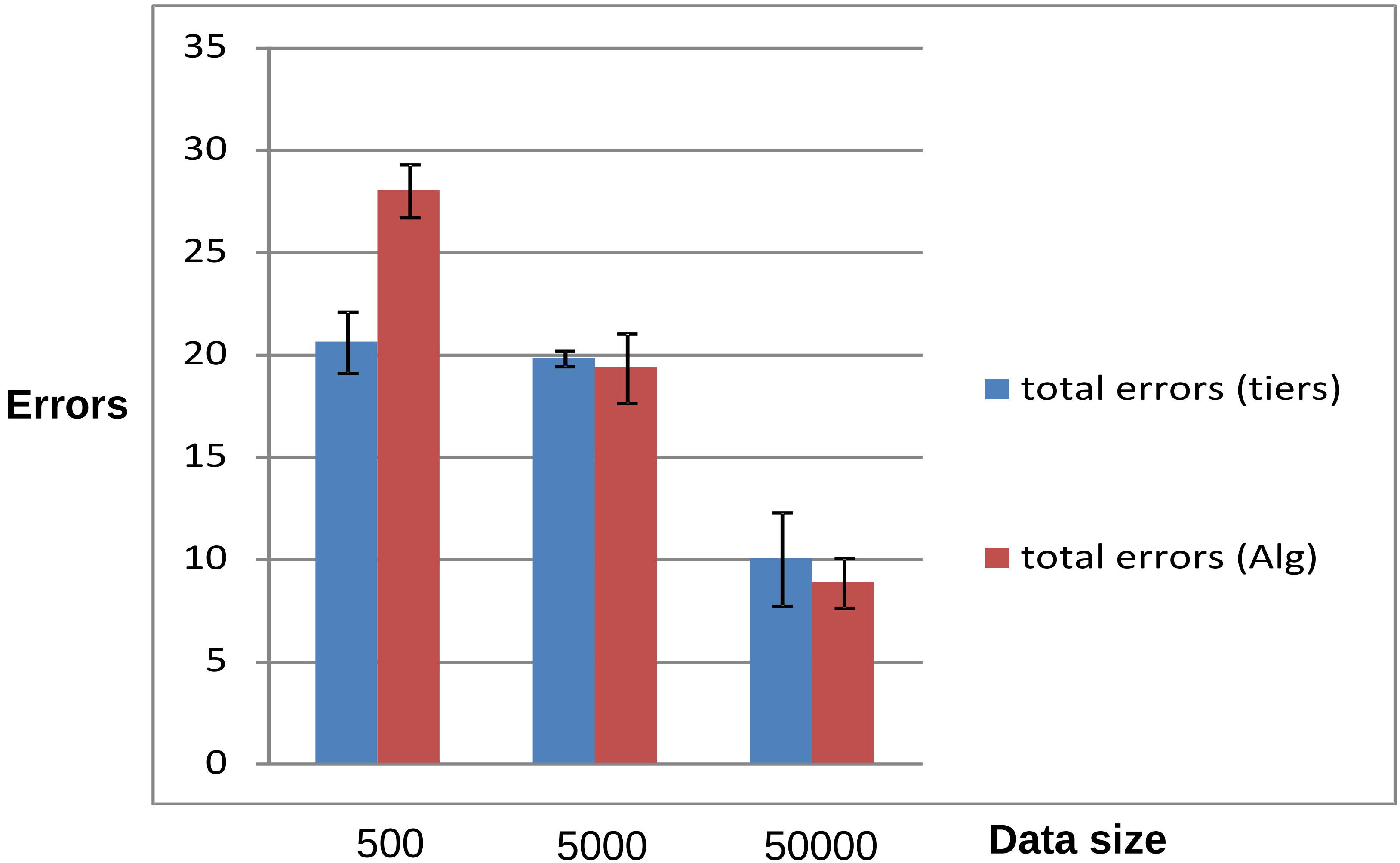
Results

Transitional arc errors for the BAT network

Datasize	CaMML w/ Tiers	GeNIe (PC)	BNT
500	6.8 (0.98)	13	7.5 (0.50)
5000	5.4 (1.62)	10	6.2 (0.74)
50000	1.0 (0.0)	10	3.7 (0.33)

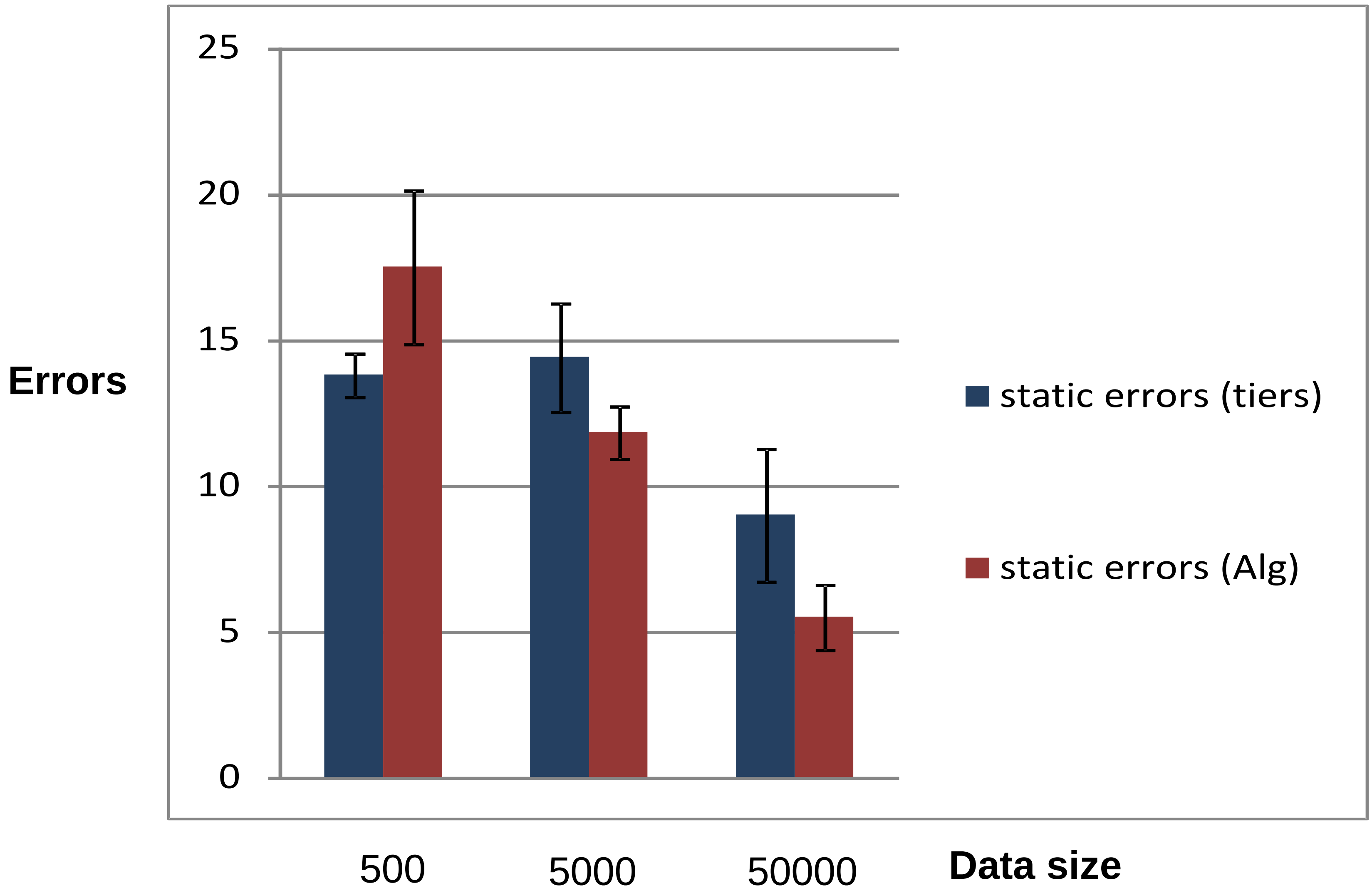
Results

BAT model (CaMML, tiers vs 2-step learning)



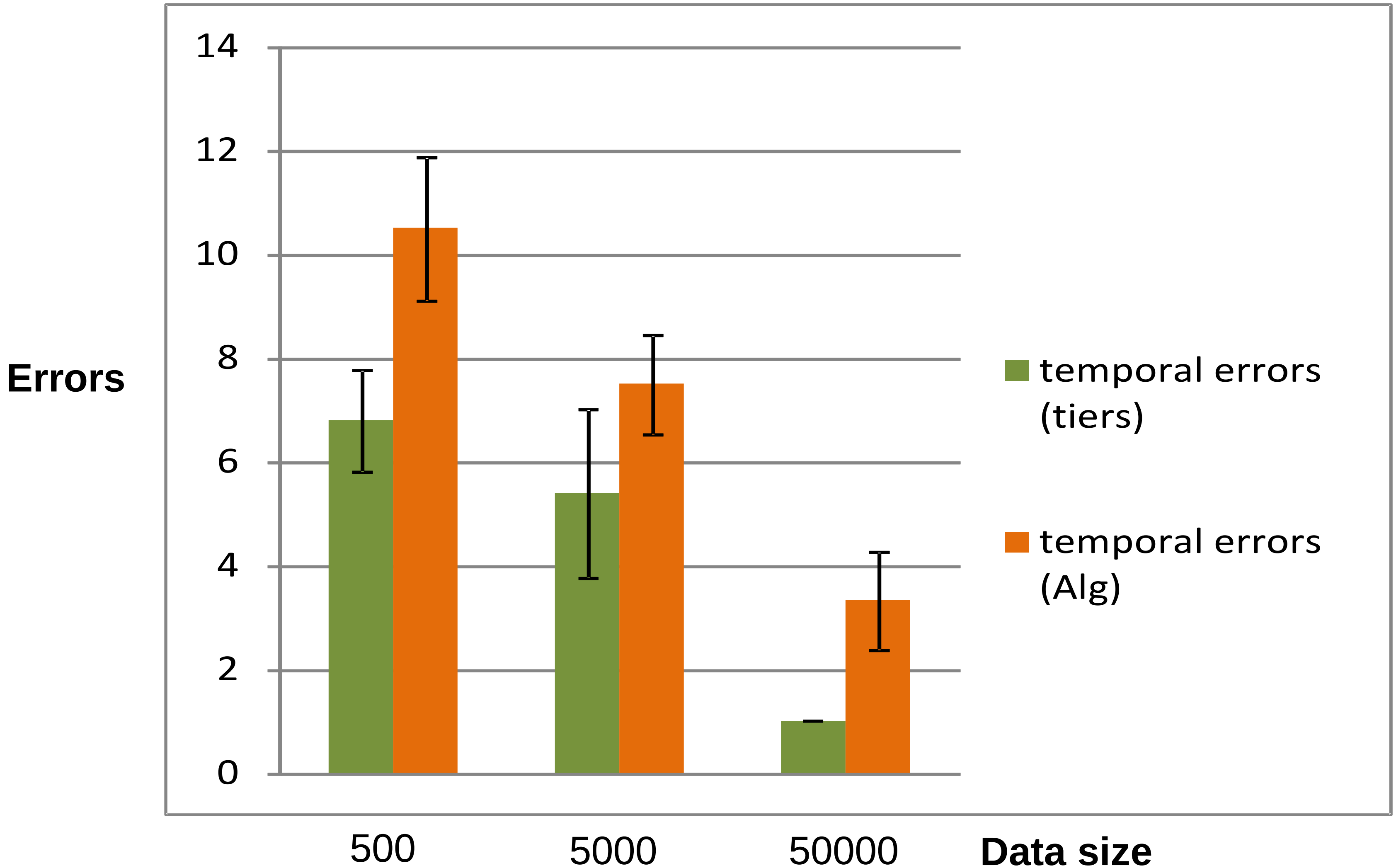
Results

BAT model (CaMML, tiers vs 2-step learning)



Results

BAT model (CaMML, tiers vs 2-step learning)



Summary

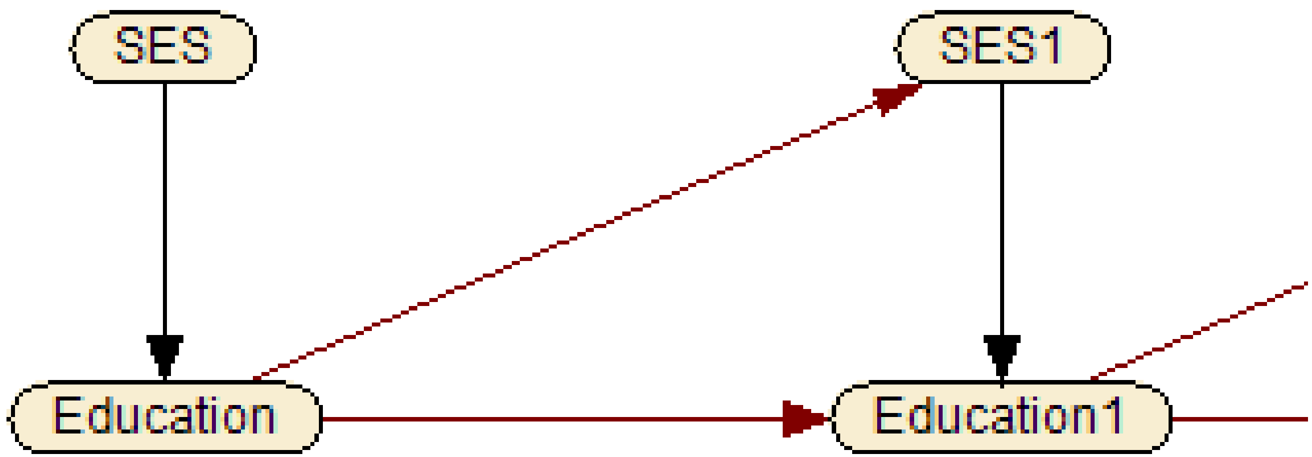
GeNIe(PC) tends to over-fit (i.e. more arcs added) with large data size in Experiment #1.

Using tiers, CaMML produces fewer errors than BNT and GeNIe(PC).

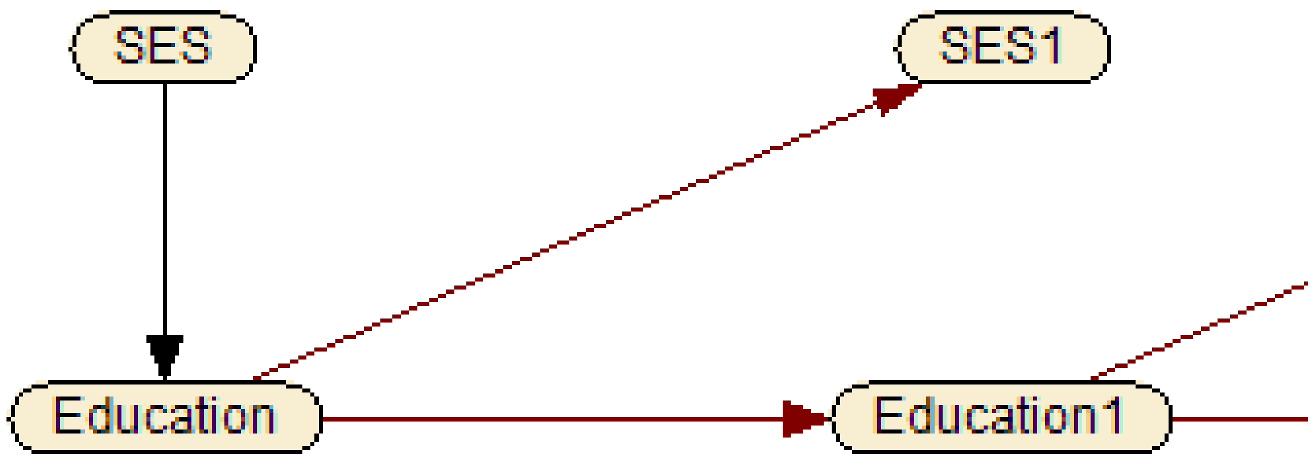
CaMML can recover more weak arcs, and usually learns all the strong arcs.

The 2-step learning algorithm produces comparable results, better at learning static arcs.

CaMML 2-Step Learning Issues

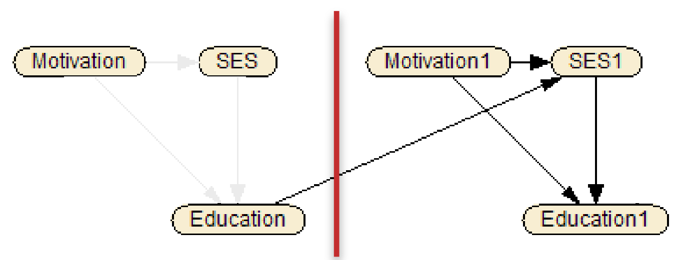


CaMML 2-Step Learning Issues



Current and Future Work

- Modify CaMML's search and score:
  - Alter score to avoid double counting static arcs



  - Alter search to avoid invalid DBN structures
  - **Ultimately:** Reduce the search space so that we can find good models more quickly