HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Partial Order MCMC for Structure Discovery in Bayesian Networks

Teppo Niinimäki, Pekka Parviainen,
Mikko Koivisto

November 25, 2011

University of Helsinki
Department of Computer Science

Outline

Introduction

Bayesian structure learning

Partial Order MCMC

Empirical results

Outline

Introduction

Bayesian structure learning

Partial Order MCMC

Empirical results

Motivation

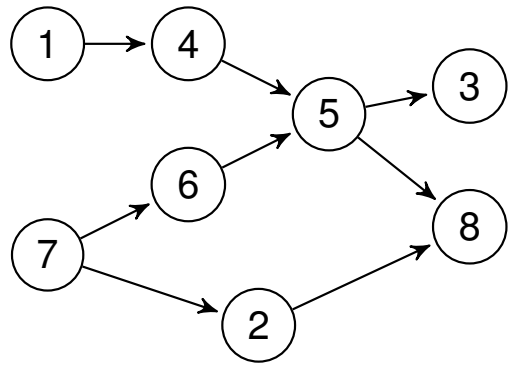
Situation: Given data *D*

	variables							
sample	1	2	3	4	5	6	7	8
1	2	1	0	1	2	2	2	1
2	2	0	2	2	0	2	2	0
3	2	0	1	1	1	1	1	0
4	1	0	2	1	2	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
5000	2	2	1	2	0	2	0	1

Task: Find relations between variables

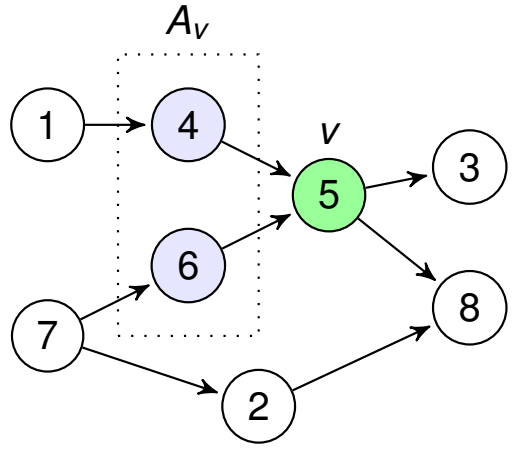
Bayesian network

- directed acyclic graph $G = (N, A)$



Bayesian network

- directed acyclic graph $G = (N, A)$



- conditional probabilities

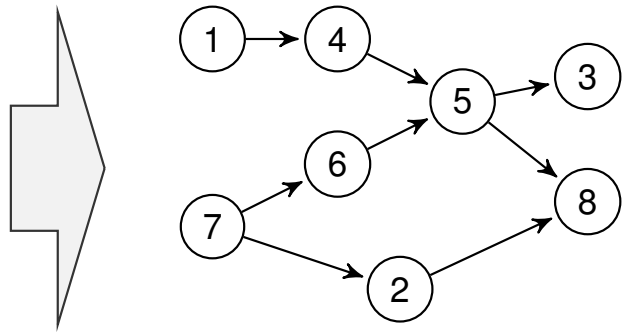
$$\Pr(D|A) = \prod_{v \in N} \Pr(D_v | D_{A_v}, A)$$

Structure learning

In general

Goal: Given data, learn the structure

	variables				
sample	1	2	3	···	8
1	2	1	0	···	1
2	2	0	2	···	0
3	2	0	1	···	0
4	1	0	2	···	1
⋮	⋮	⋮	⋮	⋮	⋮
5000	2	2	1	···	1

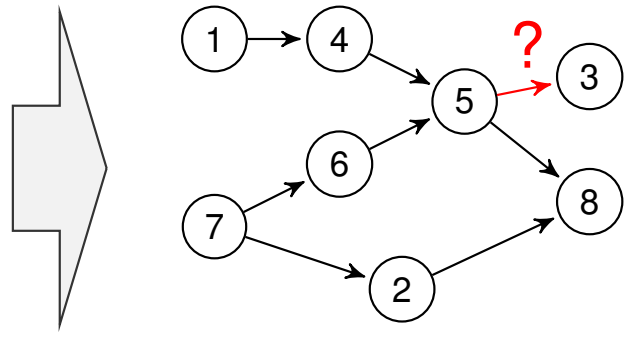


Structure learning

In general

Goal: Given data, learn the structure **or** part of it

	variables				
sample	1	2	3	···	8
1	2	1	0	···	1
2	2	0	2	···	0
3	2	0	1	···	0
4	1	0	2	···	1
⋮	⋮	⋮	⋮	⋮	⋮
5000	2	2	1	···	1



Structure learning

Approaches

Goal: Given data, learn the structure or part of it

Two approaches:

- (pairwise) independency tests
- (global) score
  - max score (MAP)
  - Bayesian averaging

Structure learning

Approaches

Goal: Given data, learn the structure or part of it

Two approaches:

- (pairwise) independency tests
- (global) score
  - max score (MAP)
  - Bayesian averaging

Outline

Introduction

Bayesian structure learning

Partial Order MCMC

Empirical results

Bayesian structure learning

Problem

Input:

- prior over structures $\Pr(A)$
- data D and likelihood $\Pr(D|A)$
- binary feature: $f(A)$

$$\text{Example: } f(A) = \begin{cases} 1 & \text{if } A \text{ contains arc } 5 \rightarrow 3 \\ 0 & \text{otherwise} \end{cases}$$

Output:

- probability $\Pr(f|D)$

Bayesian structure learning

Solution

1. Sum over structures (F&K, 2003):

$$\Pr(f|D) = \sum_A f(A) \Pr(A|D)$$

2. Use Bayes' theorem:

$$\Pr(A|D) = \frac{\Pr(D|A) \Pr(A)}{\Pr(D)}$$

Bayesian structure learning

Solution

1. Sum over structures (F&K, 2003):

$$\Pr(f|D) = \sum_A f(A) \Pr(A|D)$$

2. Use Bayes' theorem:

$$\Pr(A|D) = \frac{\Pr(D|A) \Pr(A)}{\Pr(D)}$$

$$\Pr(A|D) \propto \Pr(D|A) \Pr(A)$$

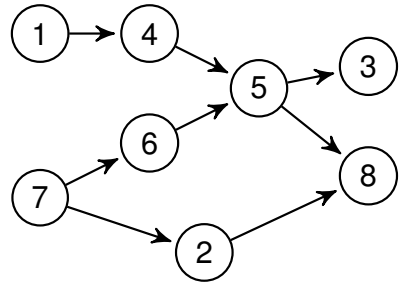
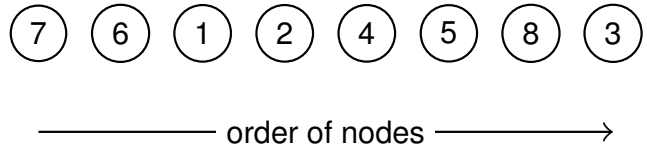
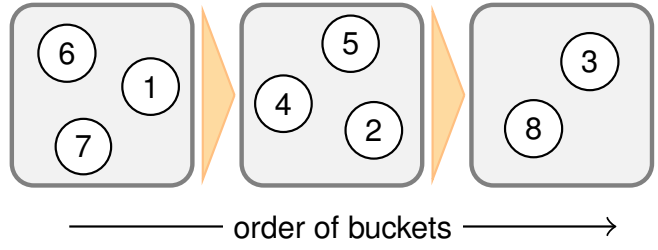
The state of the art

Method	Speed	Accuracy	OM-prior
Approximation			
■ structure MCMC (Madigan & York, 1995)	very fast	bad	No
■ linear order MCMC (Friedman & Koller, 2003)	fast	better	Yes
Exact			
■ dynamic programming (Koivisto & Sood, 2004)	slow	exact	Yes

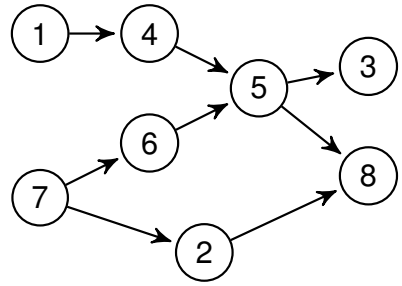
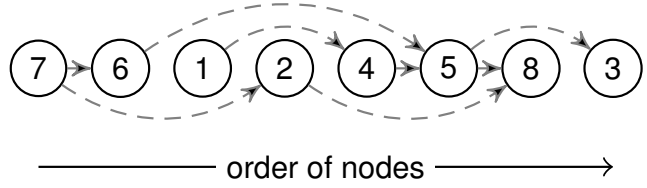
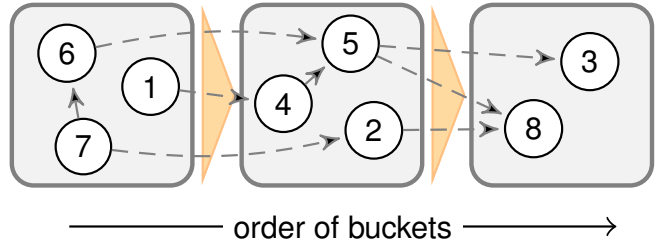
The state of the art

Method	Speed	Accuracy	OM-prior
Approximation			
structure MCMC (Madigan & York, 1995)	very fast	bad	No
linear order MCMC (Friedman & Koller, 2003)	fast	better	Yes
partial order MCMC (Niinimäki et al., 2011)	**fast**	**even better**	**Yes**
Exact			
dynamic programming (Koivisto & Sood, 2004)	slow	exact	Yes

Node orders

structure	A	
linear order	L	
partial order (bucket order)	P	

Node orders

structure	*A*	
linear order	*L*	
partial order (bucket order)	*P*	

Partition by node orders

Structure:

$$\Pr(f|D) = \sum_{A} f(A) \Pr(A|D)$$

Partition by node orders

Structure:

$$\Pr(f|D) = \sum_{A} f(A) \Pr(A|D)$$

Linear order:

$$\Pr(f|D) = \sum_{L} \Pr(f|L, D) \Pr(L|D)$$

Partial order:

$$\Pr(f|D) = \sum_{P} \Pr(f|P, D) \Pr(P|D)$$

Outline

Introduction

Bayesian structure learning

Partial Order MCMC

Empirical results

Partial Order MCMC

The algorithm:

1. Sample partial orders $P_1, \ldots, P_T$ from $\Pr(P|D)$.
2. Estimate $\Pr(f|D) \approx \frac{1}{T} \sum_{i=1}^{T} \Pr(f|D, P_i)$.

Partial Order MCMC

The algorithm:

1. Sample partial orders $P_1, \ldots, P_T$ from $\Pr(P|D)$.
2. Estimate $\Pr(f|D) \approx \frac{1}{T} \sum_{i=1}^{T} \Pr(f|D, P_i)$.

Above

$$\Pr(f|D, P_i) = \frac{\Pr(f, P_i, D)}{\Pr(P_i, D)}$$

and

$$\Pr(P|D) \propto \Pr(P, D).$$

Partial Order MCMC

The algorithm:

1. Sample partial orders $P_1, \ldots, P_T$ from $\Pr(P|D)$.
2. Estimate $\Pr(f|D) \approx \frac{1}{T} \sum_{i=1}^{T} \Pr(f|D, P_i)$.

Above

$$\Pr(f|D, P_i) = \frac{\Pr(f, P_i, D)}{\Pr(1, P_i, D)}$$

and

$$\Pr(P|D) \propto \Pr(1, P, D).$$

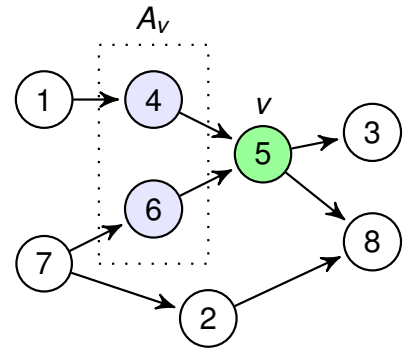
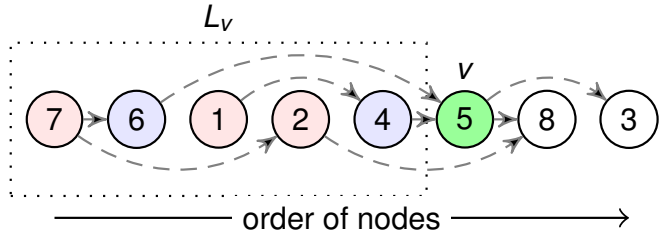
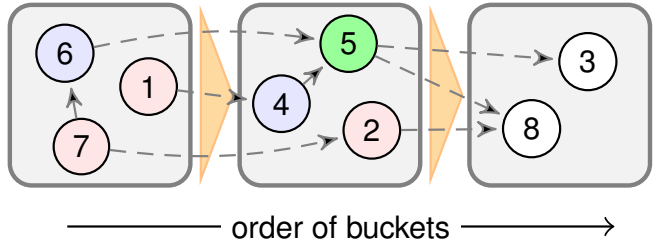
Partial Order MCMC

The algorithm:

1. Sample partial orders $P_1, \ldots, P_T$ from $\Pr(P|D)$.
2. Estimate $\Pr(f|D) \approx \frac{1}{T} \sum_{i=1}^{T} \Pr(f|D, P_i)$.

Computing $\Pr(f, P, D)$	
Sampling from $\Pr(P|D)$	

Node orders

structure	A	
linear order	L	
partial order (bucket order)	P	

Modularity assumptions

Joint probability (known):

$$\Pr(D|A) = \prod_{v \in N} \Pr(D_v | D_{A_v}, A)$$

Order-modular prior (assumed):

$$\Pr(A, L) = \prod_{v \in N} \rho_v(L_v) q_v(A_v)$$

Modular feature (assumed):

$$f(A) = \prod_{v \in N} f_v(A_v)$$

Probability for linear order

Local scores:

$$\beta_v(A_v) = q_v(A_v)\,\Pr(D_v|D_{A_v}, A) f_v(A_v)$$

Now for linear order L:

$$\Pr(f, L, D) = \prod_{v \in N} \rho_v(L_v) \sum_{A_v \subseteq L_v} \beta_v(A_v)$$

Probability for linear order

Local scores:

$$\beta_v(A_v) = q_v(A_v)\Pr(D_v|D_{A_v}, A)f_v(A_v)$$

Now for linear order L:

$$\Pr(f, L, D) = \prod_{v \in N} \rho_v(L_v) \sum_{A_v \subseteq L_v} \beta_v(A_v)$$

One more assumption: $|A_v| \leq k$
(that is, $\beta_v(A_v) = 0$ otherwise)

$$\Rightarrow \quad \Pr(f, L, D) = \prod_{v \in N} \rho_v(L_v) \sum_{\substack{A_v \subseteq L_v \\ |A_v| \leq k}} \beta_v(A_v)$$

Probability for partial order

For partial order P:

$$\Pr(f, P, D) = \sum_{L \supseteq P} \prod_{v \in N} \underbrace{\rho_v(L_v) \sum_{\substack{A_v \subseteq L_v \\ |A_v| \leq k}} \beta_v(A_v)}_{\alpha_v(L_v)}$$

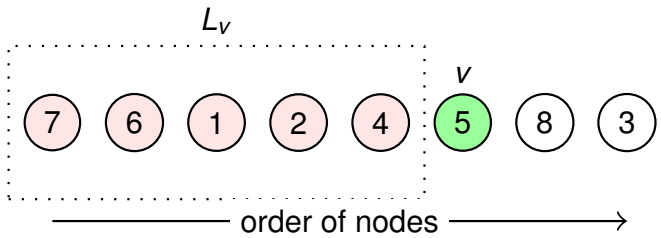
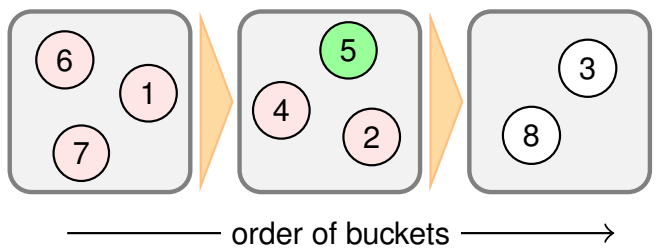
Above $L_v \in \mathcal{I}(P)$, where

$\mathcal{I}(P) =$ the set of *ideals* of P.

Probability for partial order

For partial order P:

$$\Pr(f, P, D) = \sum_{L \supseteq P} \prod_{v \in N} \underbrace{\rho_v(L_v) \sum_{\substack{A_v \subseteq L_v \\ |A_v| \leq k}} \beta_v(A_v)}_{\alpha_v(L_v)}$$

L		$\lvert\mathcal{I}(L)\rvert = n + 1$
P		$\lvert\mathcal{I}(P)\rvert = \frac{n}{b}(2^b - 1) + 1$ (b = bucket size)

Computation

Task 1: For $v \in N$ and $L_v \in \mathcal{I}(P)$, compute

$$\alpha_v(L_v) = \rho_v(L_v) \sum_{A_v \subseteq L_v} \beta_v(A_v)$$

Task 2: Compute

$$\Pr(f, P, D) = \sum_{L \supseteq P} \prod_{v \in N} \alpha_v(L_v)$$

Computation

Task 1: For $v \in N$ and $L_v \in \mathcal{I}(P)$, compute

$$\alpha_v(L_v) = \rho_v(L_v) \sum_{A_v \subseteq L_v} \beta_v(A_v)$$

$\Rightarrow$ fast sparse zeta transform, $O(n^{k+1} + n^2|\mathcal{I}(P)|)$

Task 2: Compute

$$\Pr(f, P, D) = \sum_{L \supseteq P} \prod_{v \in N} \alpha_v(L_v)$$

$\Rightarrow$ dynamic programming, $O(n|\mathcal{I}(P)|)$

Partial Order MCMC

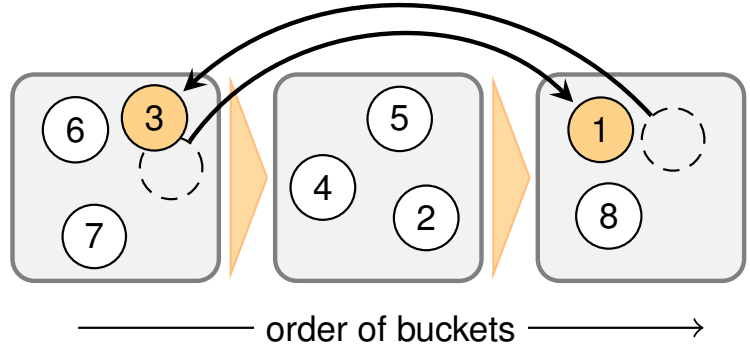
The algorithm:

1. Sample partial orders $P_1, \ldots, P_T$ from $\Pr(P|D)$.
2. Estimate $\Pr(f|D) \approx \frac{1}{T} \sum_{i=1}^{T} \Pr(f|D, P_i)$.

Computing $\Pr(f, P, D)$	$O(n^{k+1} + n^2 \lvert\mathcal{I}(P)\rvert)$
Sampling from $\Pr(P|D)$	

MCMC sampling

Sampling: Metropolis–Hastings with swaps

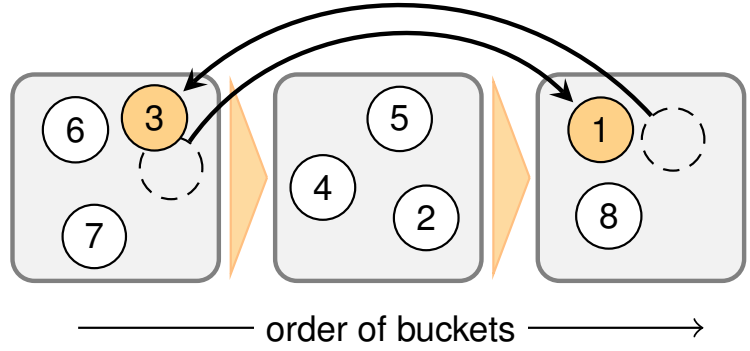


Proposal: $P \to P'$

Acceptance probability: $\min\left\{1, \frac{\Pr(P'|D)}{\Pr(P|D)}\right\}$

MCMC sampling

Sampling: Metropolis–Hastings with swaps



Proposal: $P \to P'$

Acceptance probability: $\min\left\{1, \frac{\Pr(1,P',D)}{\Pr(1,P,D)}\right\}$

Partial Order MCMC

The algorithm:

1. Sample partial orders $P_1, \ldots, P_T$ from $\Pr(P|D)$.
2. Estimate $\Pr(f|D) \approx \frac{1}{T} \sum_{i=1}^{T} \Pr(f|D, P_i)$.

Computing $\Pr(f, P, D)$	$O(n^{k+1} + n^2 \lvert\mathcal{I}(P)\rvert)$
Sampling from $\Pr(P|D)$	$O(n^{k+1} + n^2 \lvert\mathcal{I}(P)\rvert)$

Total time per sample:

$O(n^{k+1} + n^2 \lvert\mathcal{I}(P)\rvert)$

or $O(n^{k+1})$ if P is ”thin“ enough

The state of the art

Method	Time/step	States
Approximation		
structure MCMC (Madigan & York, 1995)	$O(n)$	a lot
linear order MCMC (Friedman & Koller, 2003)	$O(n^{k+1})$	$n!$
partial order MCMC (Niinimäki et al., 2011)	$O(n^{k+1} + n^2\|\mathcal{I}(P)\|)$	$1 \ldots n!$
Exact		
dynamic programming (Koivisto & Sood, 2004)	$O(n^{k+1} + n2^n)$	1

Outline

Introduction

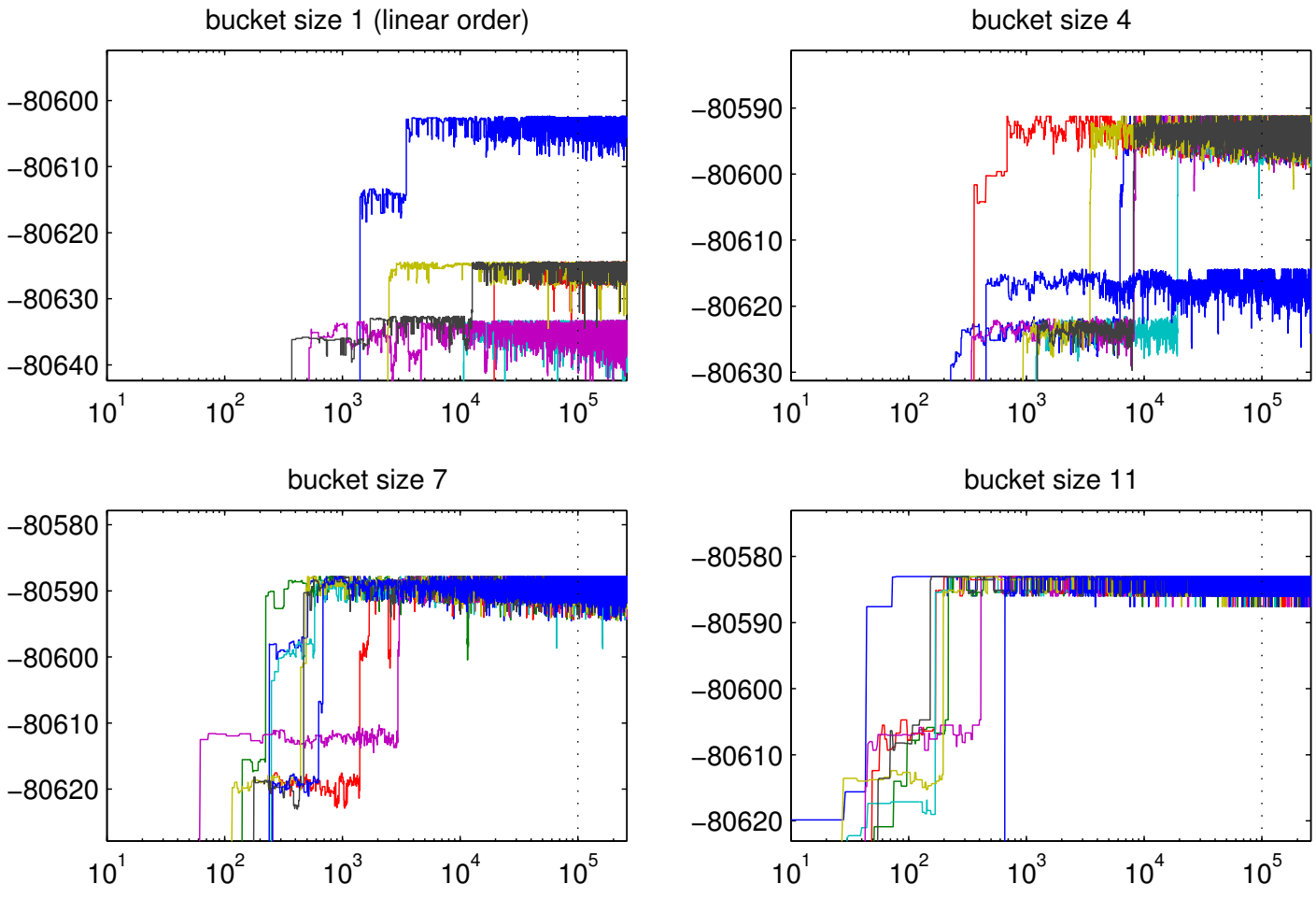
Bayesian structure learning

Partial Order MCMC

Empirical results

Empirical results: Mushroom

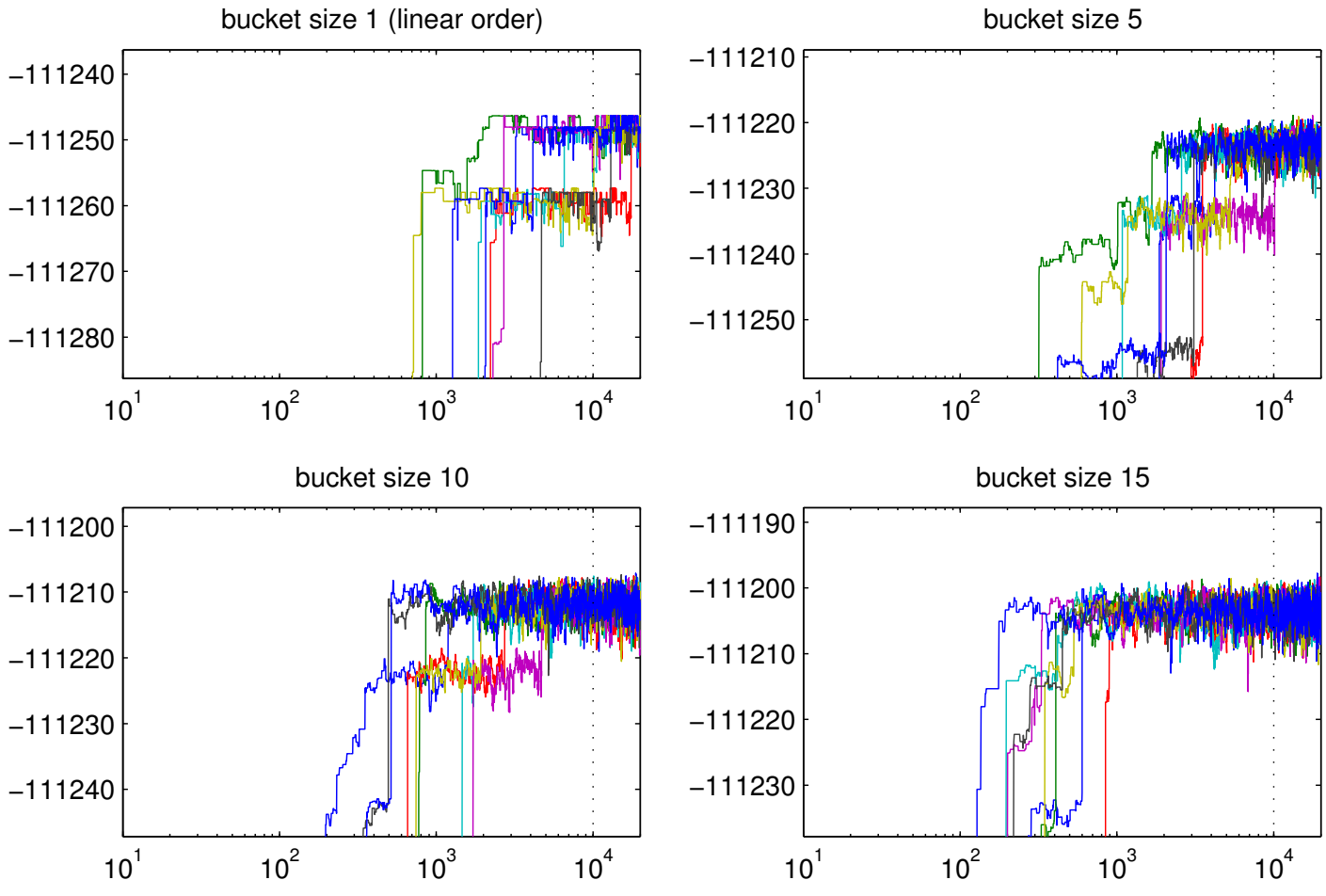
Convergence



(Mushroom-dataset, 22 variables, 8124 rows, $k = 5$)

Empirical results: Alarm

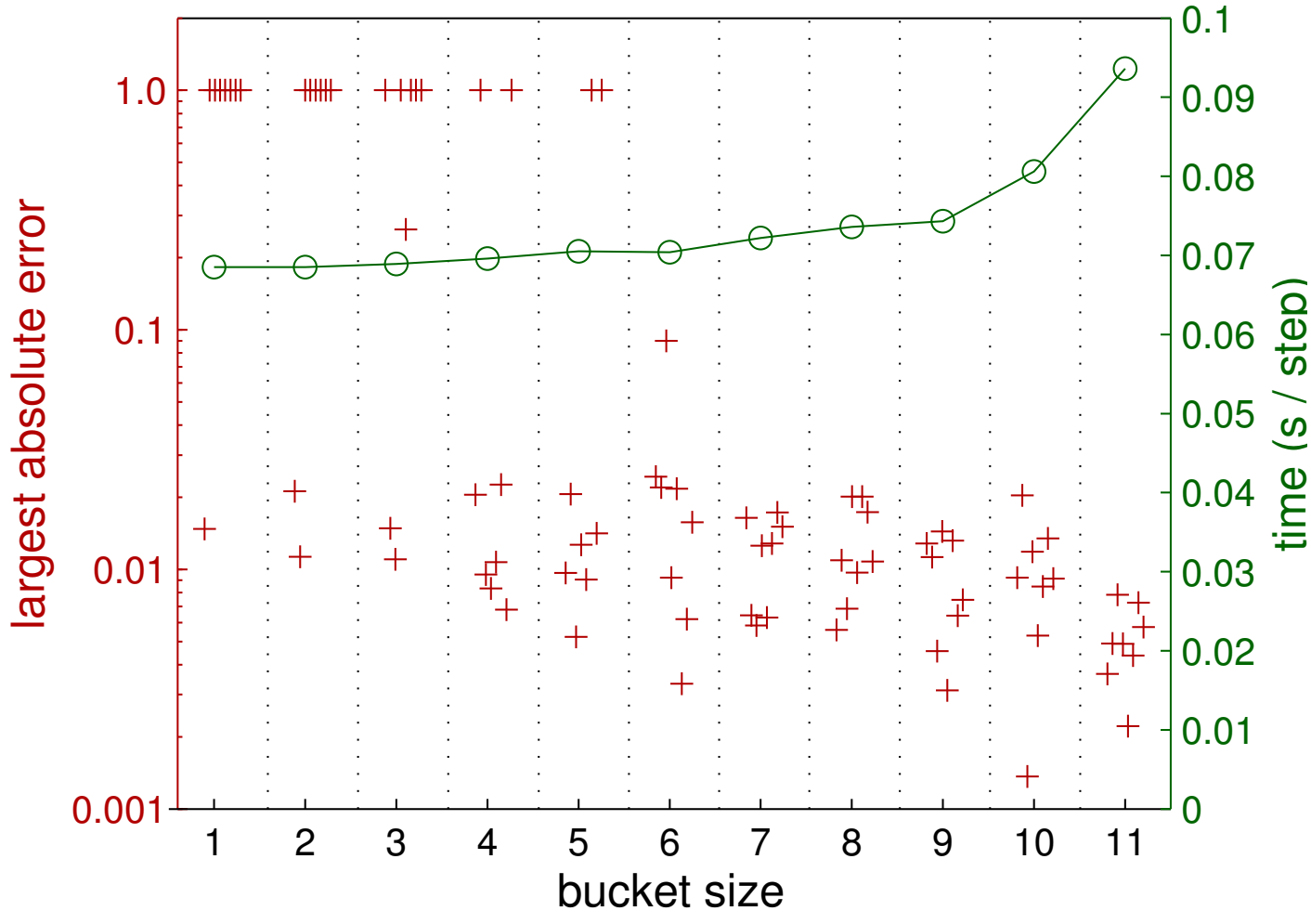
Convergence



(Alarm-network, 37 nodes, 10000 generated rows, $k = 4$)

Empirical results: Mushroom

Accuracy and time



Conclusion

Partial Order MCMC

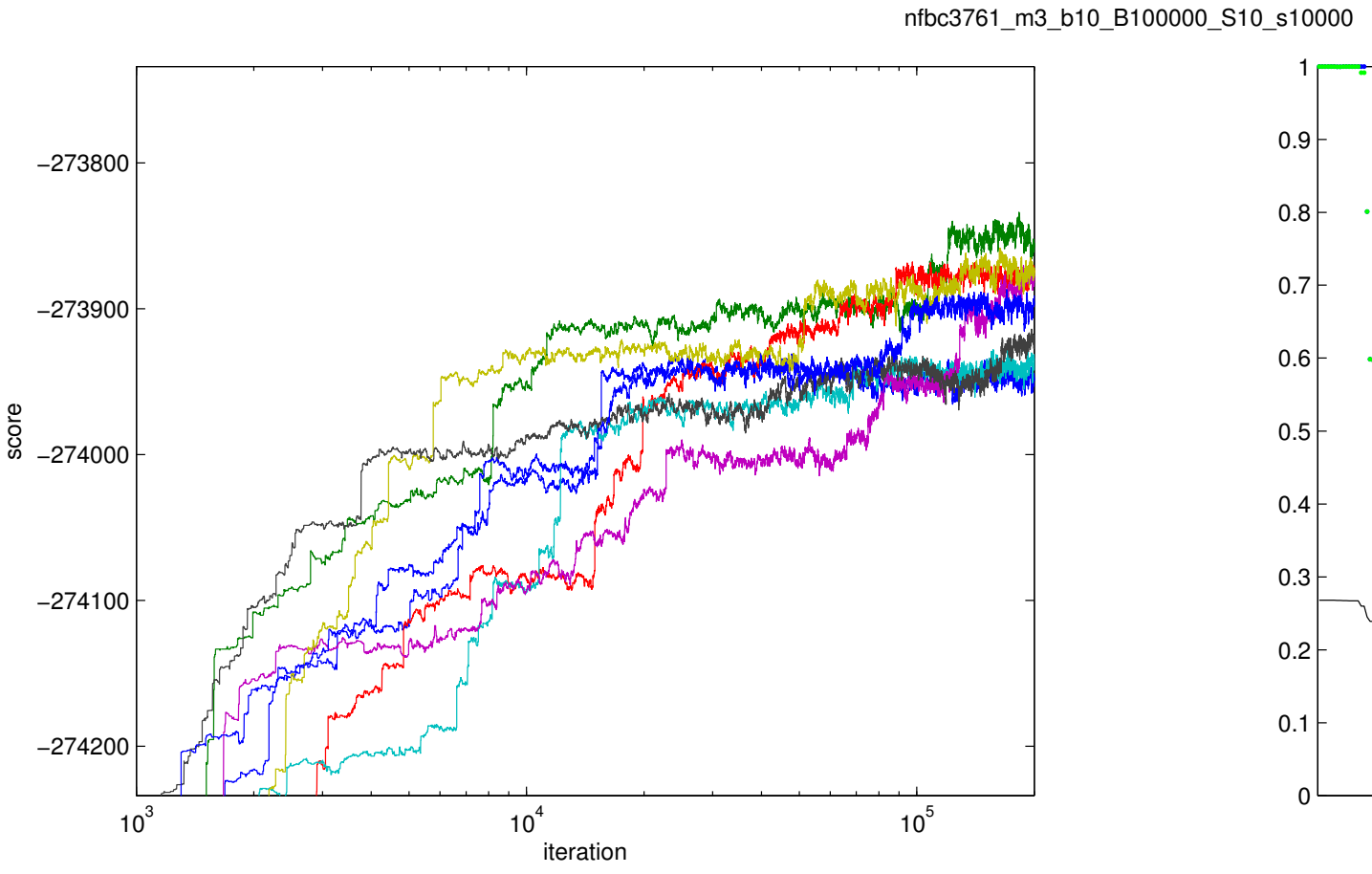
- smaller sample space, better mixing
- small increase in time per sample
- implementation available at
  `www.cs.helsinki.fi/u/tzniinim/BEANDisco`

Open problems and future work:

- improving time to $O(n^k + n^2|\mathcal{I}(P)|)$
- adaptive parent set size?
- application to real large datasets

Empirical results: NFBC

Convergence



(NFBC-data, 146 variables, 3761 rows, $k = 3$)