Layering-MCMC for Structure Learning in Bayesian Networks
Supplement

Jussi Viinikka
Department of Computer Science
University of Helsinki
jussi.viinikka@helsinki.fi

Mikko Koivisto
Department of Computer Science
University of Helsinki
mikko.koivisto@helsinki.fi

A PROOF OF LEMMA 1

We prove that $g_0(\emptyset, \emptyset) = \pi(B)$. Recall that here $B = B_1 B_2 \cdots B_\ell$ is a fixed M-layering.

Let $T \subseteq D \subseteq B_j$ and $U = B_{1:j-1} \cup D$. We show that

$$g_j(U, T) = \sum_R \prod_{i=2}^{k} f(U \cup R_{1:i-1}, R_{i-1}, R_i)\,, \qquad (1)$$

where $R = R_1 R_2 \cdots R_k$ runs through all ordered partitions of $T \cup (V \setminus U)$ such that $R_1 = T$ and that R is compatible with the layering B. Then the claim follows by Equation (2) of the main paper.

We proceed by induction on $|U|$. If $|U| = n$, then $D = B_k$, and by definition, $g_k(U, T) = 1$. This equals (1) as the sum has a single term ($R = R_1 = T$) and the empty product evaluates to 1.

Suppose then that $|U| < n$ and that the claim holds for all larger sets. We branch into two cases.

Case $D = B_j$ with $j < \ell$. Now, if $|B_{j+1}| > M$, then by definition and the induction hypothesis, $g_j(U, T)$ equals

$$f(U, T, B_{j+1}) \sum_R \prod_{i=2}^{k} f(B_{1:j+1} \cup R_{1:i-1}, R_{i-1}, R_i)\,,$$

where $R = R_1 R_2 \cdots R_k$ runs through all ordered partitions of $B_{j+1} \cup (V \setminus B_{1:j+1})$ such that $R_1 = B_{j+1}$ and that R is compatible with B. By writing $R'_2 := B_{j+1}$ and renaming $R'_{i+1} := R_i$ for $i \geq 2$, we get that $g_j(U, T)$ equals

$$\sum_{R'} \prod_{i=2} f(U \cup R'_{1:i-1}, R'_{i-1}, R'_i)\,,$$

where $R' = R'_1 R'_2 \cdots R'_k$ runs through all ordered partitions of $T \cup (V \setminus U)$ such that $R'_1 = T$ and that R' is compatible with the layering B (since we must have $R'_2 = B_{j+1}$).

Otherwise, $|B_{j+1}| \leq M$ and by definition and the induction hypothesis, $g_j(U, T)$ equals

$$\sum_{\substack{\emptyset \subset S \subseteq B_{j+1} \\ j=0 \text{ or } |S| > M - |B_j|}} f(U, T, S) \times \sum_R \prod_{i=2}^{k} f(U \cup S \cup R_{1:i-1}, R_{i-1}, R_i)\,,$$

where $R = R_1 R_2 \cdots R_k$ runs through all ordered partitions of $S \cup (V \setminus (U \cup S))$ such that $R_1 = S$ and that R is compatible with the layering B. By renaming $R'_2 := S$ and $R'_{i+1} := R_i$ for $i \geq 2$, we get that $g_j(U, T)$ equals

$$\sum_{R'} \prod_{i=2} f(U \cup R'_{1:i-1}, R'_{i-1}, R'_i)\,,$$

where $R' = R'_1 R'_2 \cdots R'_k$ runs through all ordered partitions of $T \cup (V \setminus U)$ such that $R'_1 = T$ and that R' is compatible with the layering B: indeed, if $j = 0$, there is no contraint on the size of the first part R'_2 in layer B_{j+1}, whereas if $j > 0$, it follows from the properties of M-layerings that R'_2 must not fit the previous layer, i.e., $|R_2| > M - |B_j|$.

Case $D \subset B_j$ with $j \leq \ell$. By definition and the induction hypothesis, $g_j(U, T)$ equals

$$\sum_{\emptyset \subset S \subseteq B_j \setminus U} f(U, T, S) \times \sum_R \prod_{i=2}^{k} f(U \cup S \cup R_{1:i-1}, R_{i-1}, R_i)\,,$$

where $R = R_1 R_2 \cdots R_k$ runs through all ordered partitions of $S \cup (V \setminus (U \cup S))$ such that $R_1 = S$ and that R is compatible with the layering B. By renaming $R'_2 := S$

and $R'_{i+1} := R_i$ for $i \geq 2$, we get that $g_j(U, T)$ equals

$$\sum_{R'} \prod_{i=2} f(U \cup R'_{1:i-1}, R'_{i-1}, R'_i)\,,$$

where $R' = R'_1 R'_2 \cdots R'_k$ runs through all ordered partitions of $T \cup (V \setminus U)$ such that $R'_1 = T$ and that R' is compatible with the layering B: indeed, since $D \subset B_j$, any nonempty $R'_2 \subseteq B_j \setminus D$ is a valid part in an M-layering, for R'_2 is not the first part in B_j.

This completes the proof of Lemma 1.

B GENERATING A ROOT-PARTITION

Given an M-layering B, we can generate a partition $R \in \mathcal{R}(B)$ with probability proportional to $\pi(R)$ by stochastic backtracking of the dynamic programming algorithm that computes $\pi(B)$. The pseudo code given in Figure B.1 gives one way to organize the computations.

GENERATE-PARTITION($B_1 B_2 \cdots B_\ell$)
// We assume the arrays $\hat{\tau}_j$ and g_j are available
$j = 0$; $D = \emptyset$; $T = \emptyset$
$k = 0$;
while $j \leq \ell$
  if D == B_j *// j' and D'*
    $j' = j + 1$; $D' = \emptyset$
  else $j' = j$; $D' = D$

  if $D \subset B_j$ or j == ℓ or $|B_{j+1}| \leq M$ *// $\mathcal{S}$ and A*
    $\mathcal{S} = \{S \subseteq B_{j'} \setminus D' : \emptyset \subset S\}$; $A = \emptyset$
  else $\mathcal{S} = \{B_{j+1}\}$; $A = B_{j+1} \setminus \min B_{j+1}$

  for each $v \in B_{j'} \setminus D'$ *// Construct $p[.]$*
    if T == $\emptyset$
      $p[v] = \pi_v(\emptyset)$
    else if T == B_j *// Special case*
      $p[v] = \hat{\tau}_v[\{v\}]$
    else $p[v] = \hat{\tau}_v[D] - \hat{\tau}_v[D \setminus T]$

  $f[A] = 1$ *// Construct $f[A]$*
  for each $v \in A$
    multiply $f[A]$ by $p[v]$

  draw r from $\mathrm{Unif}(0, g_j[D, T])$
  $s = 0$
  for each $S \in \mathcal{S}$ in increasing order by $|S|$
    $v = \min S$
    $f[S] = f[S \setminus \{v\}] \cdot p[v]$
    if $D \neq B_j$ or j == 0 or $|S| > M - |B_j|$
      add $f[S] \cdot g_{j'}[D' \cup S, S]$ to s
    if $s > r$ *// The next partition element is S*
      $k = k + 1$; $R_k = S$; $T = S$; $D = D' \cup S$
      break
  $j = j'$
return $R_1 R_2 \cdots R_k$

Figure B.1: Pseudo code for generating a random partition R from the conditional posterior $\pi(R|B)$ given an M-layering B. Note: $B_0 = B_{\ell+1} = \emptyset$.

C EXPERIMENTS FOR LYMPH AND HEPATITIS DATASET

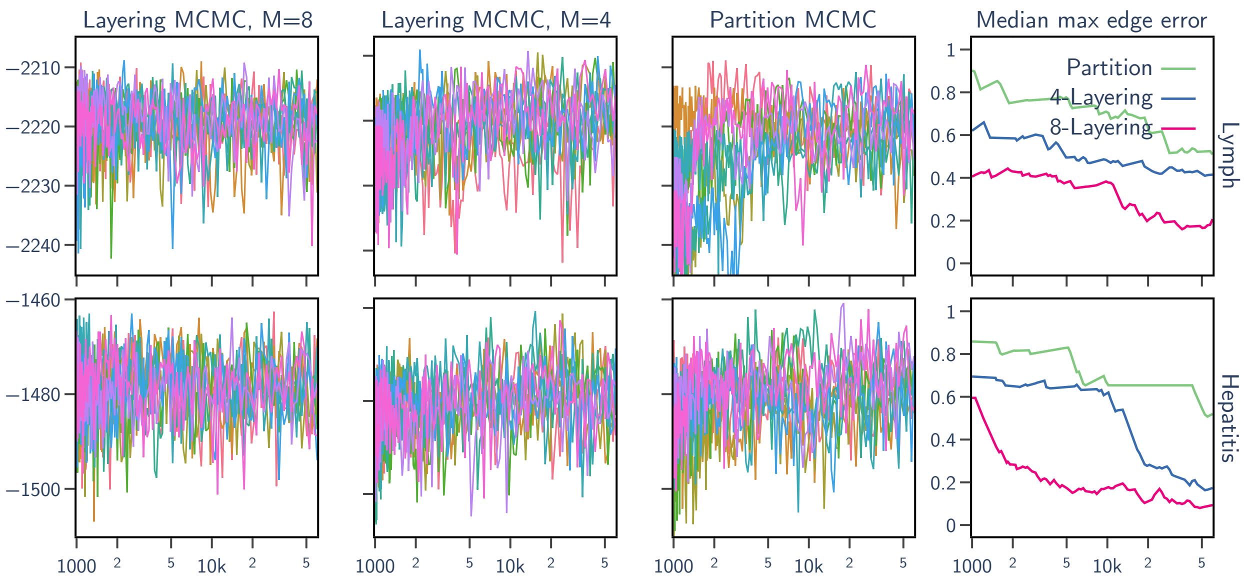

Figure C.1: Comparison of layering-MCMC and partition-MCMC on benchmark data sets. *Left:* The posterior probability of the sampled DAG (a logarithm of the unnormalized posterior) per simulation step, in nine independent runs. *Right:* The largest absolute error in the arc posterior probability estimate as a function of the length of the simulation (median over nine independent runs). Note that the x-axis is logarithmic and that, per run, shown are only 200 evenly spaced points out of the 60 000 steps.