Appendix: Sequential Graph Matching with Sequential Monte Carlo

Seong-Hwan Jun
Department of Statistics
UBC

Samuel W.K. Wong
Department of Statistics
University of Florida

James V. Zidek
Department of Statistics
UBC

Alexandre Bouchard-Côté
Department of Statistics
UBC

A Greedy Algorithm for Finding the Consensus Matching

We approximate the consensus matching by first making a pass over the particle population, $\{m^n\}_{n=1}^N$ to retrieve the edges, $e \in m^n$, as well as the count for each of the edges. We start with an empty matching, $cm = \emptyset$. The next step is to sort the edges, $\{e_{(j)}\}$, and add an edge to cm one edge at a time as long as there does not exist an edge $e' \in cm$ such that $e' \bigcap e_{(j)} \neq \emptyset$. We provide the pseudocode in Algorithm 1.

We note here that a polynomial time algorithm for finding the optimal consensus matching for bipartite graph matching can be developed using an algorithm such as Hungarian algorithm [1] (runtime complexity of $O(|\{e_{(j)}\}|^3)$). But it is unclear whether such exact algorithm can be developed for K-partite matching for $K > 2$, hence we adopt a greedy approximation in this case.

Algorithm 1 : **GreedyConsensus**($\{m^n\}$)

```
1: $H \leftarrow$ map()
2: for $n = 1, \ldots, |N|$ do
3:     for $e \in m^n$ do
4:         $H[e] \leftarrow H[e] + 1$
5:     end for
6: end for
7: $cm \leftarrow \emptyset$
8: $\{e_{(j)}\}_{j=1}^{|H|} \leftarrow$ sort($H$)
9: for $j = 1, \ldots, |\{e_{(j)}\}|$ do
10:    if $e_{(j)} \bigcap e' = \emptyset : e' \in cm$ then
11:        $cm \leftarrow cm \bigcup e_{(j)}$
12:    end if
13: end for
14: return $cm$
```

B Details on the Decision Model

The decision model is a user configurable component in our matching framework. Although it is usually a problem specific component, there are certain variations of it that can potentially affect the performance. Furthermore, one must give consideration to the overcounting problem that can arise for specific choices of the decision model. We illustrated the pairwise decision model in the main paper. Here, we present two additional decision models and provide details in regards to the overcounting problem.

B.1 Set Packing Decision Model

The set packing decision model is as follows. We visit the nodes sequentially as given by σ (random or deterministic) in the main paper. Each node decides to pack itself among the available sets. The initial matching is $m_0 = \emptyset$ and hence, the only available decision for the first node visited is to pack itself into a singleton set, i.e., $e = \{v_{\sigma(1)}\}$ and $m_1 = \{e\}$. The second node visited, assuming that it is coming from a different partition as $v_{\sigma(1)}$, has two possible decisions. The first is to grow e to $e' = \{v_{\sigma(1)}, v_{\sigma(2)}\}$ and set $m_2 = m_1 + e' = m_1 \setminus e \bigcup e'$. The second possibility is to pack itself into its own singleton set, $e' = \{v_{\sigma(2)}\}$ and set $m_2 = \{e, e'\}$. And this process continues until all of the nodes are visited.

We show that this decision model satisfies the assumptions in [2] ensuring the correctness of the SMC algorithm using this set packing decision model bypassing the need to implement the overcounting correction.

Proposition 1. *Let $\nu : \mathcal{S} \to \mathcal{S}$ be the proposal density associated with the poset $(S, \prec)$ defined as described in the main paper. Let $\pi : \mathcal{S}_R \to [0, \infty)$ be the target density defined on the state space of interest. The following are true for the SMC sampler for matching based on the set packing decision model:*

1. *$\nu^n(s \to s') > 0$ for some n if and only if $s \prec s'$, where ν^n refers to n applications of the proposal to s.*
2. *The undirected Hasse diagram corresponding to $(S, \prec)$ is connected and acyclic.*
3. *The target density π is positive for all $s \in \mathcal{S}$ and if π_* is the restriction of π to $\mathcal{M} = \mathcal{S}_R$, then, π*

extends the density π_* on $\mathcal{S}_R$ in the sense that there is a constant $C > 0$ with $\pi = C1_{\mathcal{S}_R}\pi_*$.

Proof. In our case, the target density, π, is given by the set packing decision model.

1. For $n = 1$, this is clearly true: $\nu(s \to s') > 0$ if and only if s' is in the decision set of the state s. For arbitrary $n > 1$, it follows from simple induction.

2. The Hasse diagram is acyclic if for each $s \in \mathcal{S}$, it covers at most one $s' \in \mathcal{S}$. For the initial state s_0, the empty matching, it does not cover any state so it is trivially true. For $s_r \neq s_0$, then the only state that it covers is a state that is obtained by removing the last node added, which can be obtained from the decision sequence by $d_{v_{\sigma(r)}}$. To see that the Hasse diagram is connected, note that each state $s \neq s_0$ covers exactly one state, namely the one obtained by reverting the last decision.

3. Since the sequential decision model is well defined for partial matching, this is trivially true: set $C = 1$.

□

B.2 Bipartite Graph Matching

For the general bipartite matching problem, every node in V_1 may be matched with a node in V_2 (without loss of generality, take $|V_1| \leq |V_2|$). A suitable decision model for bipartite graph matching is to visit the nodes in V_1 and consider all of the nodes in the other partition in the decision set. This is a special case of the set packing decision model and hence, we can show that it satisfies Proposition 1.

C Supplement for Sequential Monte Carlo Method

We provide a generic pseudocode for SMC for sequential graph matching in Algorithm 2.

The weight for each particle is computed as:

$$w_r^n = \alpha(s_{r-1}^{a_r^n} \to s_r^n) = \frac{\gamma_r(s_r^n)\nu^-(s_r^n \to s_{r-1}^{a_r^n})}{\gamma_{r-1}(s_{r-1}^{a_r^n})\nu^+(s_{r-1}^{a_r^n} \to s_r^n)}. \tag{1}$$

The backward kernel is as described in Section 5.4 of the main paper. The forward as well as the backward proposal depends much on the choice of the decision model. For example, for bipartite matching decision model and the set packing decision model $\nu^- = 1$ because we can ensure that there is exactly one parent state for each state. The forward proposal is carried out by formulating the decision set, $\mathcal{D}(v_{\sigma(r)})$, and sampling a decision $d_{v_{\sigma(r)}}$ exactly from Equation (4) in Section 3 of the main paper. Alternatively, one may propose to match randomly with probability $1/|\mathcal{D}(v_{\sigma(r)})|$ for faster execution as it bypasses computing the probabilities for each of the decisions. The weight update simplifies when exact forward sampling is used:

$$\alpha(s_{r-1}^{a_r^n} \to s_r^n) = \nu^-(s_r^n \to s_{r-1}^{a_r^n}). \tag{2}$$

Note that $\gamma_r(s_r^n)$ is given by the first r factors of Equation (12) in the main paper and hence,

$$\gamma_r(s_r^n)/\gamma_{r-1}(s_{r-1}^{a_r^n}) = p(d_{v_{\sigma(r)}}|m_{r-1}, \sigma_r, \theta).$$

So if exact sampling is used,

$$\nu^+(s_{r-1,a_r^n} \to s_{r,n}) = p(d_{v_{\sigma(r)}}|m_{r-1}, \sigma_r, \theta).$$

Algorithm 2 : **SMC**$(V_1, ..., V_K, N)$

```
1: $s_0^n \leftarrow \{\}$
2: $\bar{w}_0^n \leftarrow 1/N$
3: $R \leftarrow \sum_k |V_k|$
4: for $r = 1, \ldots, R$ do
5:     $a_r^n \sim$ Multinomial$(\bar{\boldsymbol{w}}_{r-1})$
6:     $s_r^n \sim \nu^+(\cdot|s_{r-1}^{a_r^n})$
7:     $w_r^n \leftarrow \alpha(s_{r-1}^{a_r^n} \to s_r^n)$
8:     $\bar{w}_r^n \leftarrow \frac{w_r^n}{\sum_{n=1}^N w_r^n}$
9: end for
10: return $(s_R^n, \bar{w}_R^n)_{n=1}^N$
```

D Additional Materials on Knot Matching

D.1 Problem Background

Common dimensions of construction lumber are the 2-by-4 and 2-by-8 (i.e., height is nominally two inches and the width of the board is 4 or 8 inches). The length of a piece varies, but our dataset is made up of boards that are 8 feet long.

One important use of lumber is for construction purposes and it is of critical importance that each piece of lumber be able to withstand certain loads. The current grading system identifies strength reducing characteristics, such as knots, which are remnants of tree branches that appear on the surfaces after sawing. In Figure 1, we have shown a sample of a board that we are using for evaluation. The four surfaces of the board are laid out side-by-side to make it clear which knots are matched with which. From Figure 1, we can see that some knots can easily be matched based on proximity.

However, there are difficult cases where there are multiple knots within close enough proximity that it would be difficult to determine the correct match without considering the sizes as well as the distance, necessitating the development of our methodology.

D.2 Features

The knot detection is carried out using a bounding box algorithm that outputs location (x, y, z) as well as the dimension (w, h) of the bounding box. Let $e = \{v_i, v_j\}$ are being considered for matching; here v_i and v_j are the nodes representing two knots. We extract a distance based feature depending and two size based features.

The distance based feature is computed as follows. If the two knots both appear on the wide surfaces, then we ignore the z coordinate in the computation of the distances and set $\phi_1(\{v_i, v_j\}) = d((v_{i,x}, v_{i,y}), (v_{j,x}, v_{j,y}))$, where $d(.,.)$ denotes the Euclidean distance. For all of the other cases, we set $\phi_2(\{v_i, v_j\}) = d((v_{i,x}, v_{i,y}, v_{i,z}), (v_{j,x}, v_{j,y}, v_{j,z}))$. The reason for separating into the distance feature into two cases is because of the preference for sawmills to cut lumber in such a way as to have the two knot faces both appear on the wide surfaces. And hence, if a knot v_i on one of the wide surfaces is presented with two candidates v_j, also on one of the wide surfaces and $v_{j'}$ on one of the narrow surfaces, we want the model to prefer it to match with v_j, all things being equal. We can achieve this by incorporating this knowledge into the prior over the parameters. What we have done is restricting the support of the parameters so that $\theta_2 < \theta_1$. Note that this does not break the convexity of the optimization procedure at hand as we have only added a linear inequality constraint $\theta_2 - \theta_1 < 0$ (refer to [3]).

The size based feature is computed by taking the absolute difference of the width and the height: $\phi_3(v_i, v_j) = |v_{i,w} - v_{j,w}|$ and $\phi_4(v_i, v_j) = |v_{i,h} - v_{j,h}|$. For the MC-EM experiments, we have restricted the parameters to be negative, $\theta_p < 0$ for $p = 1, 2, 3, 4$. This is because the feature function we defined is such that smaller value is indicative of higher probability of a match. For example, the smaller the sizes and the distance between two knots, the more likely for the two knots to be matched. Again, this is equivalent to incorporating our knowledge into the prior over the parameters and it was found to improve the convergence of MC-EM.

D.3 Monte Carlo Expectation Maximization Convergence

We have executed MC-EM for maximum of 100 iterations, with the algorithm terminating sooner if a convergence criterion on the parameters is satisfied: $\|\theta^{t+1} - \theta^t\|_1/p < \delta$ where $p = 4$ (the number of features). We have plotted the negative log likelihood versus the iterations of MC-EM in Figure 2. For these plots, $\delta = 0.01$ was chosen. We have ran the experiment 5 times for each board – the different colors indicate different replications.

For boards ID 8 and 24, we see that the MC-EM converged well before hitting the 100-th iterations. For boards 17, 18, and 20, the MC-EM reached the maximum number of iterations. For board 20, the likelihood seems stable. For boards 17 and 18, we have plotted a solid black line indicating the overall mean of the negative log likelihood across the five replications. Although the negative log likelihood does not seem to be stabilizing for boards 17 and 18, it seems to be oscillating around this mean value.

We also observed random spikes in the negative log likelihood across the figures. We have observed that these random spikes did not affect the overall performance of the MC-EM. The random spikes seem to be due to the Monte Carlo error. One remedy to handle such a problem is to increase the number of Monte Carlo samples, as the Monte Carlo approximation of the objective function is deemed to have been swamped by Monte Carlo error (see for example [4]). However, we did not implement such device as we were able to attain satisfactory performance as is (only slightly worse than supervised approach as can be seen in Table 1 of the main paper).

References

[1] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity.* Courier Corporation, 1982.

[2] A. Bouchard-Côté, A. Sankararaman, and M. I. Jordan. Phylogenetic inference via sequential Monte Carlo. *Systematic Biology*, 61(4):579–593, 2012.

[3] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[4] R. A. Levine and G. Casella. Implementations of the Monte Carlo EM algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

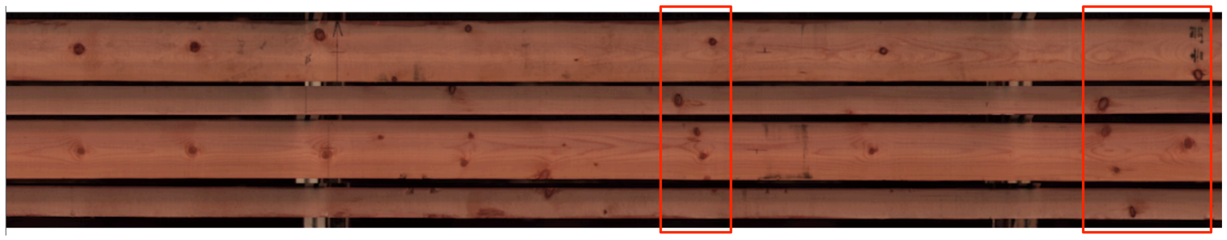

Figure 1: A sample board in our dataset. The four surfaces are laid out side-by-side. Matching that can be potentially harder are outlined inside the red box.

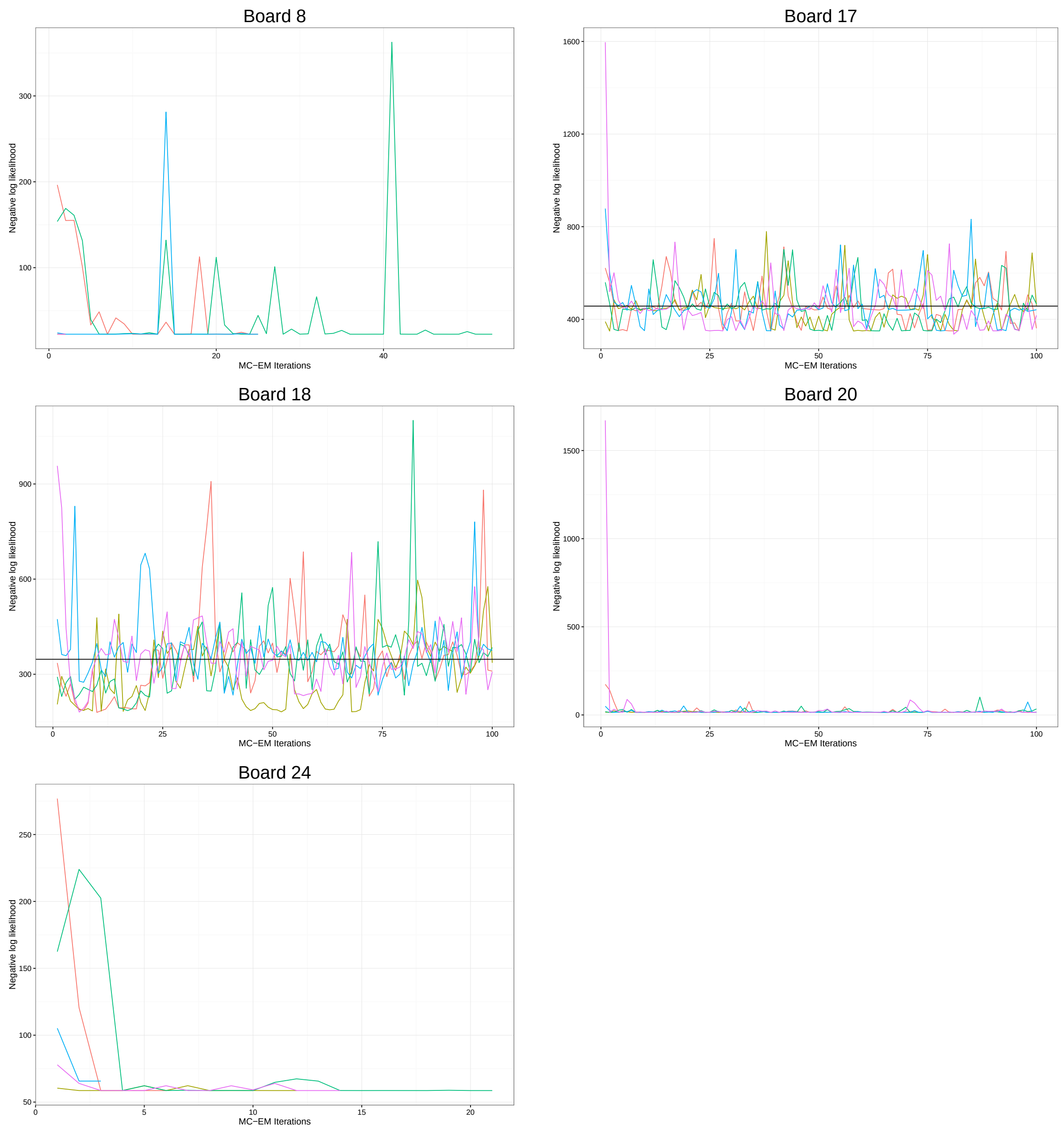

Figure 2: The negative log likelihood versus iterations of the Monte Carlo EM algorithm for 5 boards used in the knot matching experiments. The different colors indicate different runs of MC-EM.