The Annals of Applied Statistics
2019, Vol. 13, No. 3, 1678–1707
https://doi.org/10.1214/19-AOAS1255

SEQUENTIAL DECISION MODEL FOR INFERENCE AND PREDICTION ON NONUNIFORM HYPERGRAPHS WITH APPLICATION TO KNOT MATCHING FROM COMPUTATIONAL FORESTRY[1]

BY SEONG-HWAN JUN*, SAMUEL W. K. WONG†, JAMES V. ZIDEK* AND ALEXANDRE BOUCHARD-CÔTÉ*

University of British Columbia and University of Waterloo*†*

In this paper, we consider the *knot-matching* problem arising in computational forestry. The knot-matching problem is an important problem that needs to be solved to advance the state of the art in automatic strength prediction of lumber. We show that this problem can be formulated as a quadripartite matching problem and develop a sequential decision model that admits efficient parameter estimation along with a sequential Monte Carlo sampler on graph matching that can be utilized for rapid sampling of graph matching. We demonstrate the effectiveness of our methods on 30 manually annotated boards and present findings from various simulation studies to provide further evidence supporting the efficacy of our methods.

1. Introduction.

According to established rules and standards [Green, Ross and McDonald (1994)], wood processed in mills to produce sawn lumber for use in construction is assigned into grades. The grading process for a piece of lumber involves identifying its visual characteristics and assessing its strength in a nondestructive manner. The "knot," formed by a branch or limb during growth of the tree, is an important class of visual characteristics that affects both the aesthetic quality as well as the strength of wood. For individual pieces of lumber, previous studies have shown a strong relationship between the size of its knots and its strength when loaded to failure [see, e.g., Castéra, Faye and El Ouadrani (1996), Hietaniemi, Hannuksela and Silveén (2011)]. Therefore, knots have an important role in determining the grade of a piece.

Many modern mills utilize machine vision systems to automate the production process. Scanning systems incorporating lasers and cameras are used to detect the visual characteristics for quality control and for grading [Brännström (2009), Hietaniemi et al. (2014)]. Although images from such systems could be analyzed to provide detailed information about every knot on the piece, current grading rules

Received September 2018; revised February 2019.

[1] Supported by FPInnovations and a CRD grant from the Natural Sciences and Engineering Research Council of Canada.

Key words and phrases. Graph matching, sequential Monte Carlo, expectation maximization, Plackett–Luce model, computational forestry.

define standards and size limits on individual knots (or knot clusters) only. Therefore, much of the potential in the use of these systems to improve lumber strength prediction has yet to be realized. The strength-reducing effects of different knots on the piece may work together, and jointly modeling their effects may permit more accurate predictions of the ultimate strength of lumber, compared to models that consider individual knots in isolation. Toward this objective, fast and accurate algorithms for detecting and identifying all knots from surface scans of boards are needed. On that basis a new strength prediction model can be developed from the complete knot data; uncertainties in the grading and identification can be captured in a probabilistic prediction framework [see, e.g. Wong et al. (2016)].

In this paper, we consider surface scans of lumber pieces that provide images of its four long sides. In processing these images two main tasks must be performed to detect and identify knots. The first task is knot face recognition from the images; this task belongs mainly in the realm of computer vision as it shares similarities with the object recognition problem. The second task, which is the focus of this paper, is automatically identifying which of the detected knot faces on the different sides are from the same tree branch; we refer to this problem as "knot matching." Note that *knot* refers to the three dimensional convex body that is to be reconstructed from the *knot faces* that are observed on the surfaces of a piece from scanning technologies. By combining information from four-sided scans, the three-dimensional structure of the wood fibers can be characterized which is important for strength prediction [Olsson et al. (2013)].

Formally, we shall represent a piece of lumber by a quadripartite graph with each of its surfaces forming a partition, and with the knot faces as the nodes of the graph. Knot matching can thus be formulated as a quadripartite matching problem on a nonuniform hypergraph. We propose a sequential decision model to build a matching, where each decision is modeled via a local multinomial regression. This class of models is commonly used in other application areas (e.g., part-of-speech tagging in natural language processing [see, e.g., Berg-Kirkpatrick et al. (2010)]). This approach allows inference for the model parameters via maximum likelihood or maximum a posteriori estimation to be performed using standard techniques when given a sample of boards with known matchings (e.g., manually matched by a human). We then develop a sequential Monte Carlo (SMC) sampler for sampling knot matchings given the estimated parameters. The SMC sampler draws a population of particles from the space of matchings and thus also serves to estimate uncertainty in the unknown matching when applied to a future piece of lumber. We show that our SMC sampler is fast and thus permits online application for grading in lumber mills.

Thus, we anticipate that our contributions to this applied problem will enhance the sawn lumber production process in two important ways. First, each individual knot can be better assessed by capturing information from all of its visible faces much like a traditional human grader would, thereby increasing the effectiveness of automated grading without sacrificing speed and efficiency. Second, accurate automatic knot matchings will provide an important component of the necessary data for future refinement of lumber strength prediction models based on visual characteristics.

The paper is organized as follows. In Section 2 we describe the process that generates the data for knot matching. In Section 3 we introduce the graph theory notion relevant for problem formulation and provide an overview of probabilistic graph matching and related work. In Section 4 we develop a sequential decision model for constructing a matching, and in Section 5 we show that this model admits efficient parameter inference. In Section 6 we develop an SMC sampler for drawing samples from the distribution of matchings defined on a nonuniform quadripartite hypergraph and how it can be utilized for prediction. In Section 7 we present a procedure to simulate realistic knot-matching data for evaluating our model and SMC sampler. We present experimental results on simulated and real data in Section 8 and conclude the paper with a brief discussion in Section 9.

2. Data for knot matching.

As shown in Figure 1, the data necessary for the development of our application are generated for a piece of lumber. High-definition cameras are installed to capture images of the four surfaces as it moves on a carrier (i.e., each piece is taken along conveyor belt through the scanning station). Note that each piece of lumber has six sides but the two ends are typically ignored, as controlled sawing leaves no knot faces to appear on the end sides. The first processing task is to identify the *knot faces* on the boards. A sizeable knot causes noticeable grain deviations as wood fibers must travel around it to maintain their continuity. To ensure that a high accuracy of knot detection is achieved, the images are augmented with laser scans of tracheids (Daval et al. (2015), Olsson et al. (2013)) which measure the grain angles. Combining the two sources of data, we developed a knot detection algorithm that outputs information on the location and size of the knot faces. The second task, and the focus of this paper, is to identify which of the knot faces on the different surfaces belong to the same knot (i.e., from the same tree branch). In this section we provide the details of the relevant data for the knot-matching problem.

2.1. *Lumber and knot representation.*

The raw images and laser scans of tracheids are first processed by an internally developed knot detection algorithm.

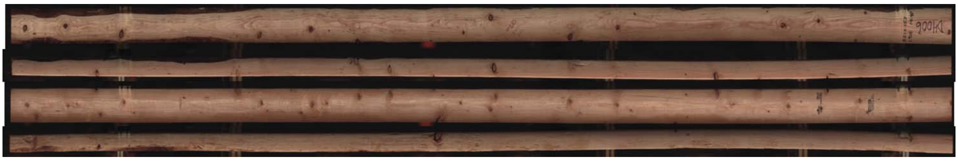


FIG. 1. *Sample lumber used in the real data analysis. Four sides of the boards with the wide surfaces shown on the first and the third rows and the narrow surfaces shown on the second and the last rows.*

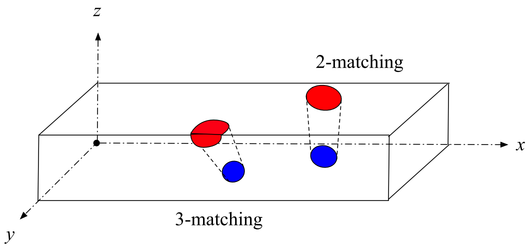


FIG. 2. *Three-dimensional view of the lumber* (*not to scale*). *An illustration of two-matching and three-matching are provided.*

Knots are typically modeled as elliptical cones [Guindos and Guaita (2013)], and hence, our implementation of the knot detection algorithm fits an ellipse over each knot face. We view each piece of lumber as a three-dimensional object, positioned in a standard three-dimensional Euclidean space as shown in Figure 2, with the x, y and z-axes representing length, width and height respectively. In this fashion the two "wide" surfaces in Figure 3(a) and (c) are parallel to the x–y plane, while the two "narrow" surfaces in Figure 3(b) and (d) are parallel to the x–z plane.

For each knot face on each surface, the knot detector outputs the three-dimensional coordinate (x, y, z), indicating the position of the center of the knot face. It also outputs the axes of the fitted ellipse on the knot face, denoted by (a, b), where a is the length of the axes along the x-axis and b is the length of the axes along the y-axis for the two "wide" surfaces and z-axis for the two "narrow" surfaces. Additionally, we have the rotation angle of the fitted ellipse, denoted α. In

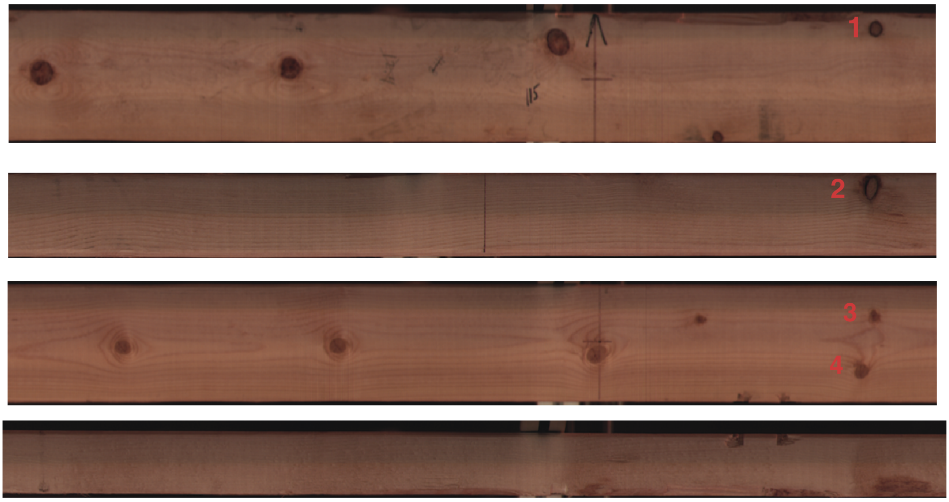


FIG. 3. *A closer look at a segment of a plank. The matching for knot faces labeled* 1, 2, 3, 4, *produced by the human grader is* $\{\{1, 4\}, \{2, 3\}\}$.

summary, each knot face is represented by the six-tuple $(p, x, y, z, a, b, \alpha)$, where p denotes the index of the surface (i.e., partition).

2.2. *Choice of covariates*. For knot faces on distinct surfaces, we can compute a vector of associated covariates to assess whether the knot faces belong to the same branch and, hence, should be matched together. Covariates that are useful predictors would help distinguish matches from nonmatches. For each pair of knot faces u and v on distinct surfaces, we considered the following covariates:

- Both u and v appear on a wide surface: We compute the Euclidean distance between u and v. We observed from our data the most common occurrence among the matched knots is of this type. Knot faces at shorter distances apart are more likely to be matches.
- One of u or v appears on a narrow surface: This covariate resembles the one above in that we compute the Euclidean distance between u and v. We found that differentiating this case from the one above helped to improve the prediction accuracy of the matching method.
- Comparison of sizes: To assess the size difference between knot faces, we compute the areas of their fitted ellipses and compute the absolute difference $|u_{\text{area}} - v_{\text{area}}|$. Knot faces belonging to the same branch are expected to have a smaller difference in sizes.

For a triplet of knot faces u, v, w, we consider the following covariates:

- Maximum and minimum distances: We compute the Euclidean distance between each pair of knot faces and extract the maximum and the minimum pairwise distances as covariates. Recall that the knot faces represent a surface of a convex body that appear when an elliptical cone is sliced. Therefore, two of the knot faces must share an axis as shown in Figure 2. However, we found that inaccuracies during the knot detection stage can potentially capture two knot faces that share an axis to appear separated. This error is of a reasonable size, and we found that computing the distance between the nearest knot faces instead is a useful approximation that leads to good empirical performance. The maximum distance is analogous to the distance covariate computed above for a pair of knot faces.
- Comparison of sizes: To adapt the size covariate above, we sum the area of the fitted ellipses of the two closest knot faces and take the absolute difference with the area of the remaining knot face.

These covariates are incorporated in the matching model developed in Section 4. We estimate the parameters associated with each of these covariates from a sample of boards where the correct knot matching is known.

2.3. *Matching by a human grader.* For a well-trained person matching the knot faces is an easy task (although time consuming). Using the visual characteristics of the knot faces to determine the matches, that human grader would examine a piece of lumber from end-to-end. For example, note the knots labeled 1 to 4 in red on Figure 3. The grader is able to determine the correct matching after careful examination. Knot face 1 and knot face 4 belong to the same branch, and knot face 2 and knot face 3 belong to the same branch. However, there are cases that can be difficult even for a human grader, and we would like to quantify uncertainty in the matching using probabilities.

We utilized a human grader to manually annotate our data. For each board each knot is represented by $(p, x, y, z, a, b, \mathit{label})$, where *label* is a unique identifier given to knot faces that stem from a same branch. The manually annotated matchings will be used to evaluate the performance of our approach.

3. Overview of graph matching.

We denote a *graph* by $G = (V, E)$, where V is a set of nodes in the graph and E denotes the set of edges. We shall require several extensions to the basic notion of a graph, namely, *hypergraph*. In a hypergraph an edge can contain one or more nodes. A hypergraph is *d-uniform* if all of the edges contain d nodes and otherwise referred to as *nonuniform*. A *K-partite* hypergraph is typically denoted by $G = (V_1, V_2, \dots, V_K, E)$, where $V_1, V_2, \dots, V_K$ are disjoint sets of nodes referred to as partition sets such that the union is equal to V and E denotes a set of edges such that each edge $e \in E$ may not contain two nodes of the same partition. The special case with $K = 2$ is known as a *bipartite* graph.

In graph theory terms each surface of the board represents a partition set, and the knot faces appearing on the surfaces correspond to the nodes. The edge set contains any combination of knot faces as long as it does not contain knot faces from the same surface. This is because a tree branch cannot manifest itself more than once on any one surface. Hence, we have the same restriction as in the K-partite graphs, where no edge can contain nodes from the same partition. Formally then, we view a piece of lumber as a complete (nonuniform) *four*-partite hypergraph.

A typical set up for a maximal graph matching problem is that given a graph $G = (V, E)$ and a weight function $w : E \to [0, \infty)$, we wish to find a matching $M \subset E$ such that $\sum_{e \in M'} w(e) \le \sum_{e \in M} w(e)$ for any other matching $M' \subset E$. Usually, this requires: (i) computing the weight function, (ii) finding high-weight matchings. We provide an overview below and develop our approaches in the following sections.

3.1. *Computing the weight function.* A common practice in machine learning is to choose a parametric model for the weight function. One common model is the *Gibbs measure* [Bouchard-Côté and Jordan (2010), Petterson et al. (2009)],

$$p(M = m|\theta) = \frac{e^{-w(m;\theta)}}{\sum_{m' \in \mathcal{M}} e^{-w(m;\theta)}} \tag{3.1}$$

which defines a probability distribution over the space of matchings. Here, we let M denote a random matching and m a realized matching, with $\mathcal{M}$ denoting the sample space of matchings for a given graph. The parameters are denoted by $\theta \in \mathbb{R}^d$, and $w(m; \theta)$ denotes the weight of the given matching m. Note that we have overloaded the notation for w to be defined on the edges as well as matchings.

Given θ, we can use equation (3.1) to compute the probability for any matching m. A linear function $w(m; \theta) = -\sum_{e \in m} \theta^T \phi(e)$ is commonly adopted as the weight function, where $\phi(e)$ denotes the covariates extracted from edge $e \in m$. This model is preferred in the machine learning community because of its *exchangeability*, that is, the order in which the edges in the matching are observed does not affect the probability of a given matching.

However, inference for the parameters based on the model in equation (3.1) can be quite challenging due to the normalization constant, whose computation requires enumeration over the space of matchings. Given a sample of known graph matchings, computing the gradient of equation (3.1) is inefficient and, hence, rules out gradient-based procedures for optimization of the parameters (e.g., to find the maximum likelihood estimate) [Petterson et al. (2009)]. For the same reason it is difficult to sample from the posterior distribution of the parameters given data. Standard Bayesian inference methods such as MCMC would encounter the so-called *doubly intractable* problem, where one would have to compute the normalization constant at each iteration of the MCMC to evaluate the Metropolis–Hastings acceptance probability [Møller et al. (2006)].

In this work we develop a novel supervised learning methodology for the model we introduced in [Jun et al. (2017)]. Our earlier work focused on unsupervised learning but relied on a heuristic approach for supervised learning. Principled adaptation of the model to the supervised context presents challenges due to the combinatorial nature of the state space. We address the challenges involved with supervised learning of K-partite matchings using a Monte Carlo expectation maximization algorithm, where sequential importance sampling with resampling algorithm is used for the Monte Carlo E-step.

3.2. *Searching for matchings.* Finding the maximal matching is a combinatorial optimization problem that has been extensively studied in the graph theory community [see, e.g., Bondy and Murty (1976), Kuhn (1955), Papadimitriou and Steiglitz (1982)]. In particular, the bipartite matching problem has received much attention due to its wide array of applicability to disciplines such as computer vision, computational biology and information retrieval among others [Caetano et al. (2009), Cao et al. (2007), Holmes and Rubin (2001), Lunter et al. (2005)]. In statistics, the problem of bipartite matching has been applied to the design of experiments when a pair of similar subjects need to be matched, and to causal inference where the goal is to match similar observational units [see, e.g., Hansen (2004), Lu and Rosenbaum (2004)]. Given the weight function, there are deterministic algorithms such as the Hungarian algorithm that can find a maximal matching in polynomial time for bipartite graphs [Kuhn (1955)]; however, such algorithms are unavailable for general K-partite graphs.

We proposed an SMC sampler for matching for K-partite graph in our earlier work [Jun et al. (2017)] by building on two important developments from sequential Monte Carlo literature. The development in Del Moral, Doucet and Jasra (2006) allows SMC to be used for inference for general state spaces and Wang, Bouchard-Côté and Doucet (2015) describes the *overcounting problem* that can arise when using SMC to explore a combinatorial space and provides theoretical results helpful toward addressing this problem. Jun et al. (2017) focuses on sampling matching for standard K-partite graphs, where the cardinality of an edge is restricted to exactly two nodes. This paper overcomes this restriction and thus offers a new sampling methodology for K-partite *hypergraphs*. By working through a unique application of knot matching, we aim to demonstrate the overcounting problem in a clear manner and provide guidelines on how to address it.

4. Sequential decision model for K-partite hypergraph matching.

In Section 4.1, we provide an expanded exposition of our earlier model proposed in Jun et al. (2017). Section 4.2 provides worked out examples on formulation of decision set for a standard bipartite graph as well as for knot-matching problem.

4.1. *Sequential decision model.*

We shall consider a matching to be represented by a sequence of decisions. That is, nodes are visited one-by-one to decide their set membership, given past decisions. Each decision in the sequence is modeled using multinomial logistic regression. Recall the Gibbs model in equation (3.1), where we noted that parameter estimation is difficult in general. In contrast our sequential representation admits efficient parameter inference.

Let $\sigma : \{1, \dots, |V|\} \to \{1, \dots, |V|\}$ be the sequence in which the nodes are visited, where V denotes the set of all nodes of a graph $G = (V, E)$. For ease of exposition, we will first assume that this permutation is known and fixed; in general this sequence can be either random or deterministic. Each node, $v_{\sigma(r)} \in V$ for $r = 1, \dots, |V|$, makes a decision, and the decision made by node $v_{\sigma(r)}$ is to be denoted by $d_{v_{\sigma(r)}}$. The set of decisions available for a node is to be denoted by $\mathcal{D}(v_{\sigma(r)}, m_{r-1})$. Here, we use m_{r-1} to denote the partial matching implied by the sequence of decisions, $\{d_{v_{\sigma(1)}}, \dots, d_{v_{\sigma(r-1)}}\}$, that is, for each decision sequence we have a mapping $\{d_{v_{\sigma(1)}}, \dots, d_{v_{\sigma(r-1)}}\} \to m_{r-1}$ where $r - 1$ nodes have made decisions. In the rest of the paper, we will often omit m_{r-1} for notational simplicity and just write $\mathcal{D}(v_{\sigma(r)})$. Interpretation for the decision set $\mathcal{D}(v_{\sigma(r)})$ is as a set of candidate edges that $v_{\sigma(r)}$ can be placed into. To be precise, we can think of making a decision d as first forming a new edge $d' = d \cup \{v_{\sigma(r)}\}$ and updating a matching by setting $m_r = m_{r-1} \setminus d \cup d'$.

The decision set is flexible and can be chosen to suit the problem at hand. For example, in bipartite matching without any restrictions it is sufficient to visit the nodes in V_1, and the decision candidate for a node $v_{\sigma(r)} \in V_1$ consists of all unmatched nodes in V_2. This can be represented by setting

$$\mathcal{D}(v_{\sigma(r)}) := \bigcup_{\substack{u \in V_2: \\ \forall v \in V_1, (u,v) \notin m_{i-1}}} \{u\}.$$

The decision set formulation also permits singleton sets where a node is placed into a set by itself which is achieved by including an empty set in the decision set. We model each decision by a multinomial regression involving the covariates extracted from an edge d', denoted $\phi(d')$, and the parameter vector θ,

$$p(d_{v_{\sigma(r)}} | m_{r-1}, \sigma, \theta) = \frac{\exp[\theta^T \phi(d_{v_{\sigma(r)}} \cup \{v_{\sigma(r)}\})]}{\sum_{j=1}^{|\mathcal{D}(v_{\sigma(r)})|} \exp[\theta^T \phi(d_j \cup \{v_{\sigma(r)}\})]}, \tag{4.1}$$

that is, each decision has a multinomial distribution with $|\mathcal{D}(v_{\sigma(r)})|$ categories. Note that we are not restricted to local covariates; it is also possible to include global features where ϕ is defined on the matching, $\phi(m_r)$.

Taking the product of the local multinomial probabilities induces the likelihood model as follows:

$$L(d_\sigma | \sigma, \theta) = \prod_{r=1}^{|V|} p(d_{v_{\sigma(r)}} | m_{r-1}, \sigma, \theta). \tag{4.2}$$

This model is akin to the Plackett–Luce model [Caron, Teh and Murphy (2014), Plackett (1975)] and commonly used for modeling ranking and preferences. One can also express the joint distribution of the decisions and the permutation as

$$p(d_\sigma, \sigma | \theta) = \prod_{r=1}^{|V|} p(d_{v_{\sigma(r)}} | m_{r-1}, \sigma_r, \theta) p(\sigma_r | \sigma_{r-1}), \tag{4.3}$$

where σ_r is a partial map $\sigma_r : \{1, \dots, r\} \to \{1, \dots, |V|\}$.

4.2. *Specification of decision models for knot matching.* In this section we provide details about the decision models we use for the knot-matching problem. For ease of exposition, we begin by providing an example of a decision model for the bipartite-matching problem and proceed to describe the specifics of a decision model that we have considered for the knot-matching application.

4.2.1. *Bipartite matching.* Bipartite matching is a special case where it is sufficient to form a matching by visiting the nodes in only one partition. An illustration of the decision set is shown in Figure 4(a). In this illustration we visit the nodes in V_1 with permutation $\sigma = \{3, 1, 5\}$. First, the decision set for node 3 is all of the nodes in V_2: $\{\{2\}, \{4\}, \{6\}\}$. From this set the node 2 is chosen (marked as yellow) first. The process continues for node 1 and then node 5 to form the bipartite matching shown in the figure.

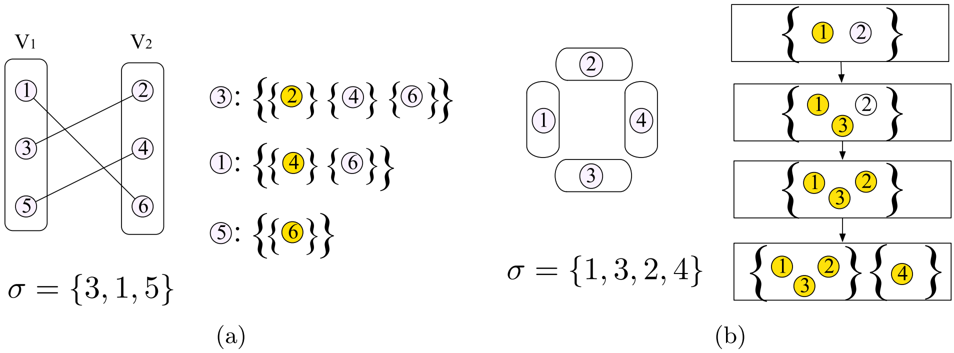
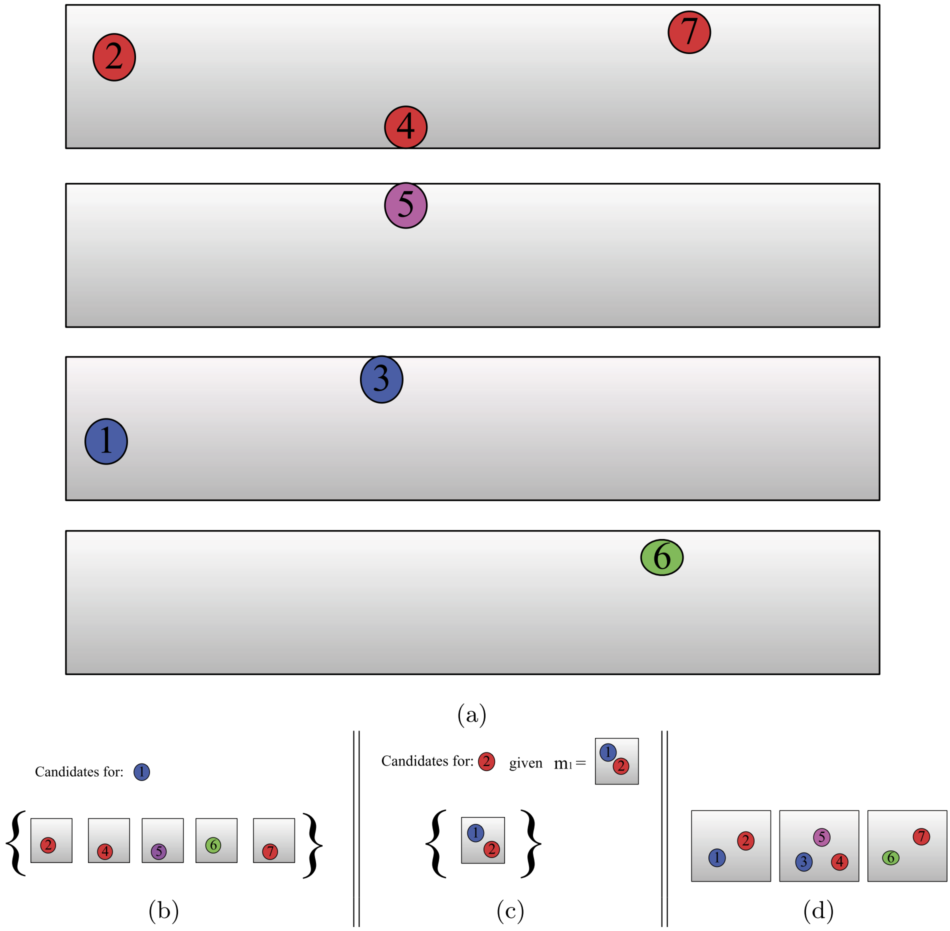


FIG. 4. (*a*) *Illustration of the decisions for a bipartite matching. For bipartite matching we only need to visit the nodes in one of the partitions. In this illustration we have* $\sigma = (3, 1, 5)$. *The decision set for the nodes are presented with the selected nodes colored in yellow.* (*b*) *An illustration of sequential decision model used for knot-matching application with each partition containing exactly one knot, labeled from* 1 *to* 4. *The rectangles represent matching, and curly braces represent edges. The visited nodes are colored in yellow whereas the nodes that are yet to be visited are unfilled.*

4.2.2. *Knot matching.* For the knot-matching problem we begin by imposing a restriction on the cardinality of the edges in the matching to be restricted to $\{2, 3\}$. This restriction stems from the fact that 4-matching is rarely observed in practice. With this restriction in place, a decision set for an uncovered knot v can be formulated to include any knot face on a different surface from v as well as any edge whose cardinality is 2 and does not contain a knot face from the same partition as v. If v is already covered (belongs to an edge), we formulate the decision set with only an empty set, equivalent to allowing no decisions. We have provided an illustration of a knot-matching problem in Figure 5(a) and illustrative decision sets in Figure 5(b) and (c), with $\sigma = \{1, 2, 3, \dots, 7\}$, prespecified. In Figure 5(b) we have $m_0 = \varnothing$, and, hence, it considers all of the nodes in the graph with different colors as candidates. In Figure 5(c) we note that the red node labeled #2 is already contained in an edge, and, hence, the only decision available is an empty set (for illustration purposes, we have indicated the edge that contains it in the decision set). Finally, an example of a final matching state is provided in Figure 5(d).

Note that this decision model allows a singleton set as a by-product. For example, consider the case with one node in each partition shown in Figure 4(b). Given a permutation $\sigma = (1, 3, 2, 4)$, the decision set for node 1 is $\{2, 3, 4\}$. Suppose it matches with node 2. Node 3 is presented with decision set $\{\{1, 2\}, \{4\}\}$. Suppose it decides to form $\{1, 2, 3\}$. Next, we note that node 2 is already covered, so it is presented with an empty set as the only decision. Then, when we visit node 4, the only decision presented to it is an empty set because the edge $\{1, 2, 3\}$ is already saturated. Therefore, node #4 forms a singleton set. In practice a singleton case may arise due to an imperfect knot detection step.



(a)

(b)

(c)

(d)

FIG. 5. (*a*) 4*-partite hypergraph representing a piece of lumber.* (*b*) *The decision set for blue node #*1. (*c*) *The decision set for red node #2 given the decision made by blue node #*1. (*d*) *An example of a final matching. This is a modified version of a similar figure appearing in Jun et al.* (2017).

5. Parameter estimation via Monte Carlo expectation maximization. With the model in place, it remains to address the problem of estimating its parameters. We focus on the supervised learning set up, where we are given a data set of I matchings: $m^1, \dots, m^I$. As noted in Jun et al. (2017), in principle this can be approached by sampling a permutation σ from its posterior distribution and followed by sampling a sequence of decisions that yield the observed matching. However, the posterior distribution over σ is intractable. Jun et al. (2017) proposed a heuristic that consists in instead sampling σ from the uniform distribution over matchings. Empirically, this approach works well when a large number of training instances is available, as was the case for image matching application considered in Jun et al. (2017). Here, we develop an inference method based on Monte Carlo expectation maximization with better understood theoretical properties and describe a sequential importance sampling (SIS) algorithm with resampling for performing the Monte Carlo E-step [Doucet and Johansen (2011)].

We would like to maximize $p(\theta|m^1,\dots,m^I)$. One potential difficulty is that for a given matching m^i, there can be multiple paths (i.e., permutation and decision sequences) that lead to m^i. For example, consider the bipartite matching $\{\{1,6\},\{3,2\},\{5,4\}\}$ (shown in Figure 4(a)). This matching can be attained with $\sigma=(1,3,5)$ where $d_{\sigma(1)}=\{6\}$, $d_{\sigma(2)}=\{2\}$, and $d_{\sigma(3)}=\{4\}$ as well as $\sigma=(3,1,5)$ where $d_{\sigma(1)}=\{2\}$, $d_{\sigma(2)}=\{6\}$, and $d_{\sigma(3)}=\{4\}$.

We view the permutation and the decisions as latent variables and express the complete data likelihood as

$$\prod_{i=1}^{I} L_c(m^i,\sigma^i,\boldsymbol{d}_{\sigma^i}|\theta)=\prod_{i=1}^{I} p(m^i|\sigma^i,\boldsymbol{d}_{\sigma^i})p(\sigma^i,\boldsymbol{d}_{\sigma^i}|\theta) \tag{5.1}$$

$$=\prod_{i=1}^{I} 1[(\sigma^i,\boldsymbol{d}_{\sigma^i})\to m^i]\times p(\sigma^i,\boldsymbol{d}_{\sigma^i}|\theta). \tag{5.2}$$

Note that $p(\sigma^i,\boldsymbol{d}_{\sigma^i}|\theta)$ is given by equation (4.3), and the $1[(\sigma^i,\boldsymbol{d}_{\sigma^i})\to m^i]$ is an indicator function equal to 1 if and only if $(\sigma^i,\boldsymbol{d}_{\sigma^i})$ maps to m^i. The inference can be carried out iteratively using the expectation maximization [Dempster, Laird and Rubin (1977)]:

$$\text{(E step): } Q(\theta,\theta^t)=\mathbb{E}[\log p(\theta|\{m^i,\sigma^i,\boldsymbol{d}_{\sigma^i}\}_{i=1}^{I})],$$
$$\text{(M step): } \theta^{t+1}=\operatorname*{argmax}_{\theta} Q(\theta,\theta^t),$$

where expectation is taken with respect to $(\sigma^i,\boldsymbol{d}_{\sigma^i})\sim p(\sigma,\boldsymbol{d}_\sigma|\theta^t,m^i)$. Note that the posterior can be expressed as

$$p(\theta|\{m^i,\sigma^i,\boldsymbol{d}_{\sigma^i}\}_{i=1}^{I})\propto\prod_{i=1}^{I} L_c(m^i,\sigma^i,\boldsymbol{d}_{\sigma^i}|\theta)p(\theta).$$

Therefore, the Q function can be expressed as

$$\begin{aligned} Q(\theta,\theta^t)&\propto\sum_{i=1}^{I}\mathbb{E}[\log L_c(m^i,\sigma^i,\boldsymbol{d}_{\sigma^i})]+\log p(\theta)\\ &=\sum_{i=1}^{I}\sum_{\sigma^i,\boldsymbol{d}_{\sigma^i}}\log L_c(m^i,\sigma^i,\boldsymbol{d}_{\sigma^i})p(\sigma^i,\boldsymbol{d}_{\sigma^i}|\theta^t,m^i)+\log p(\theta).\end{aligned} \tag{5.3}$$

We use a Monte Carlo version of EM to approximate the expectation involved in the E-step [Wei and Tanner (1990)]. To sample from $p(\sigma,\boldsymbol{d}_\sigma|\theta^t,m^i)$, we use sequential importance sampling with the state space at each iteration of the SMC as $\mathcal{S}_r = \Sigma_r \times \mathcal{D}_r$, where Σ_r is the set of all possible permutation sequences of length r, and $\mathcal{D}_r$ is the set of all possible decision sequences of length r. We will let $\sigma_r \in \Sigma_r$ denote the partial map $\sigma_r : \{1, \dots, r\} \to \{1, \dots, |V|\}$. The intermediate target distribution for iteration r is

$$p(\sigma_r^i, \boldsymbol{d}_{\sigma_r^i} | \theta^t, m^i) = \frac{1[(\sigma_r^i, \boldsymbol{d}_{\sigma_r^i}) \in m^i] p(\sigma_r^i, \boldsymbol{d}_{\sigma_r^i} | \theta^t)}{p(m^i | \theta^t)}.$$

We use the notation $(\sigma_r^i, \boldsymbol{d}_{\sigma_r^i}) \in m^i$ to mean the following. If $(\sigma_r^i, \boldsymbol{d}_{\sigma_r^i}) \to m_r^i$ is such that each $e \in m_r^i$ is contained in some edge $e \in m^i$, then we say $(\sigma_r^i, \boldsymbol{d}_{\sigma_r^i}) \in m^i$.

The proposal distribution we use at iteration r is $p(\sigma_r, \boldsymbol{d}_{\sigma_r} | \theta^t)$, in which case the weight update for proposing $(\sigma_{r+1}^i, \boldsymbol{d}_{\sigma_{r+1}^i})$ from $(\sigma_r^i, \boldsymbol{d}_{\sigma_r^i})$ is

$$\alpha\big((\sigma_r^i, \boldsymbol{d}_{\sigma_r^i}) \to (\sigma_{r+1}^i, \boldsymbol{d}_{\sigma_{r+1}^i})\big) = 1\big[(\sigma_{r+1}^i, \boldsymbol{d}_{\sigma_{r+1}^i}) \in m^i\big].$$

The proposal step is followed by resampling step using multinomial distribution defined on the normalized weights, which allows to prune the proposed decisions with zero weights, that is, when $(\sigma_r^i, \boldsymbol{d}_{\sigma_r^i}) \notin m^i$.

With the target and the proposal distribution clearly defined, we can sample the latent permutation and the decisions to approximate the Q function in equation (5.3),

$$\tilde{Q}(\theta, \theta^t) = \sum_{i=1}^{I} \frac{1}{N} \sum_{n=1}^{N} \log L_c\big(m^i, \sigma^{i,n}, \boldsymbol{d}_{\sigma^{i,n}}\big) + \log p(\theta), \tag{5.4}$$

where $(\sigma^{i,n}, \boldsymbol{d}_{\sigma^{i,n}}) \sim p(\sigma^i, \boldsymbol{d}_{\sigma^i} | \theta^t, m^i)$ for $n = 1, \dots, N$ for each $i = 1, \dots, I$.

The M-step can be carried out using numerical optimization procedures. Since each decision is modeled using a multinomial logistic, the likelihood admits exact computation of the gradient. If the gradient can be computed exactly for $p(\theta)$, then efficient numerical optimization of the objective function over the parameters using off-the-shelf optimization routines, such as L-BFGS [Liu and Nocedal (1989)], can be adopted. For example, if we take the isotropic Gaussian prior over θ, then the objective function is

$$\tilde{Q}(\theta, \theta^t) = \sum_{i=1}^{I} \frac{1}{N} \sum_{n=1}^{N} \log L_c\big(m^i, \sigma^{i,n}, \boldsymbol{d}_{\sigma^{i,n}}\big) - \lambda \|\theta\|^2, \tag{5.5}$$

for some $\lambda > 0$.

6. SMC sampler for matching and prediction. In this section, given the MAP estimate of the parameters (obtained by following the procedure described in Section 5), we describe SMC sampler for matching. Furthermore, we describe how the samples can be used for prediction. We base the exposition in this section on two important developments from the SMC literature. The first is the SMC samplers method [Del Moral, Doucet and Jasra (2006)] which extends basic SMC by introducing a sequence of intermediate distributions such that the SMC algorithm is defined on the common state space with the final distribution coinciding with the desired target distribution. This idea allows one to draw samples from an arbitrary state space. The second is the combinatorial SMC [Wang, Bouchard-Côté and Doucet (2015)] which establishes theoretical conditions for obtaining consistent estimator for SMC algorithms operating on combinatorial state spaces.

6.1. *Background and notation.* We begin by establishing notation for defining intermediate target distributions as well as intermediate state spaces. The state space of interest is the space of matchings on G, which we denote by $\mathcal{M}$. We generalize this space and introduce $\mathcal{M}_r, r = 1, \ldots, R$ as the intermediate state spaces. The state space $\mathcal{M}_r$ denotes a matching that can be realized after r nodes have made decisions using our sequential decision model. We use r to index the iterations of the SMC algorithm; therefore, R is equal to the total number of nodes in the graph to be visited. This leads to $\mathcal{M}_R = \mathcal{M}$, the space where every node has made a decision and, hence, placed into an edge.

The intermediate distributions will be denoted by γ_r and the proposal distribution by ν^+. We will use N to denote the number of particles in the SMC population and denote by $s_{r,n}$ and $w_{r,n}$ the particle n at iteration r and its un-normalized weight. The resampling step of the SMC algorithm is carried out using the normalized version of the weights, $\bar{w}_{r,n} = w_{r,n} / \sum_{n=1}^{N} w_{r,n}$. Note that the resampling step of the SMC algorithm induces the notion of a parent particle for each particle; we denote the index of the particle $s_{r,n}$ by a_r^n. The weight computation is carried out in a recursive manner,

$$w_{r,n} = \bar{w}_{r-1,a_r^n} \times \alpha(s_{r-1,a_r^n}, s_{r,n}),$$

where $\bar{w}_{r-1,a_r^n} = 1/N$ if resampling is carried out in the previous iteration of SMC and $\alpha(s_{r-1,a_r^n}, s_{r,n})$ is the weight function:

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

The particles and weights at the final iteration are used to approximate expectations of functions $f : \mathcal{M} \to \mathbb{R}$ via

$$\mathbb{E}[f(M)] \approx \frac{1}{N} \sum_{n=1}^{N} f(s_{R,n}),$$

if resampling is carried out after the last iteration and

$$\mathbb{E}[f(M)] \approx \sum_{n=1}^{N} \bar{w}_{R,n} f(s_{R,n}),$$

if resampling is not performed at the last iteration. We refer to Doucet and Johansen (2011) for an excellent exposition of SMC methods.

6.2. *Partially ordered set.* An important notion that we need to introduce before we can complete the specification of our SMC sampler is that of a partially ordered set and how it arises in the SMC setting.

A partial order $\prec$ defined on a set $\mathcal{S}$ is a binary relation that is reflexive, antisymmetric and transitive (often denoted as a pair, $(\mathcal{S}, \prec)$). The difference between a total order $<$ and the partial order $\prec$ is that not all elements of $\mathcal{S}$ are required to be comparable. That is, there may exist elements $s, s' \in \mathcal{S}$ such that neither $s \prec s'$ nor $s \succ s'$. The notion $s = s'$ for the partial orders is the same as for the total orders, that is, $s = s'$ if and only if $s \prec s'$ and $s \succ s'$. We introduce the notion of Hasse diagram of a partially ordered set [Bouchard-Côté, Sankararaman and Jordan (2012)]:

DEFINITION 1. For $s, s' \in \mathcal{S}$, s' *covers* s if $s \prec s'$ and there does not exist $s'' \in \mathcal{S}$ such that $s \prec s'' \prec s'$.

DEFINITION 2. The Hasse diagram on $(\mathcal{S}, \prec)$ is an undirected graph $G = (\mathcal{S}, E)$ where the nodes of the graph are the elements of $\mathcal{S}$ and there is an edge between the nodes $s, s' \in \mathcal{S}$ if and only if s covers s'.

In the context of the sequential Monte Carlo sampler, the notion of a partial order on the state space $\mathcal{S}$ is characterized by the proposal distribution, s covers s' if s' can be obtained by one application of the proposal to s. In other words we view proposal distribution as extending $s \in \mathcal{M}_r$ to $s' \in \mathcal{M}_{r+1}$. An example of a Hasse diagram corresponding to the decision model for the bipartite graph given in Section 4.2.1 is shown in Figure 6(a). We have also provided an example of a case where $s \prec s'$ on the top panel of Figure 6(b) and a case where two states are not comparable in the bottom panel of Figure 6(b). In essence the sequential structure that is needed by the SMC method is induced by the partial order.

The next natural question concerns the conditions for a valid proposal distribution that ensures correctness of SMC algorithm. This is provided in [Wang, Bouchard-Côté and Doucet (2015)] for combinatorial state spaces. One condition that is of great importance is that of *connectedness*, that is, starting from an initial state s_0, one should be able to reach any other state $s \in \mathcal{S}$ by finite number of applications of the proposal on s_0.

6.3. *SMC sampler for graph matching.* In Section 5 we used SMC for sampling the latent variables $(\sigma, \boldsymbol{d})$. Our goal in this section differs in the sense that the object of interest includes matching as well as permutation and the decision sequences. In this section we develop an SMC sampler that operates on an expanded state space that admits the sampling of matchings. First, recall that

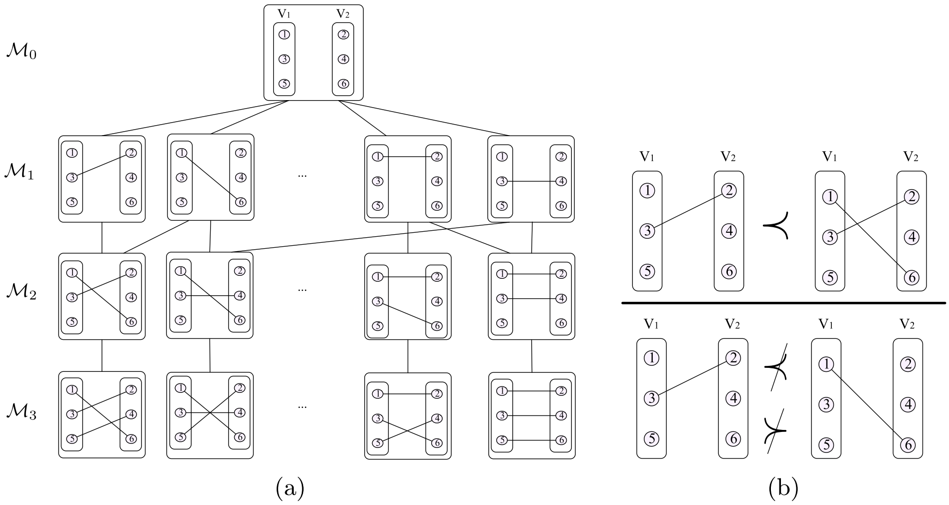


FIG. 6. (*a*) *Example of Hasse diagram corresponding to the bipartite decision model described in Section* 4.2.1. (*b*) *Example of partial order defined on bipartite matching.*

$(\sigma_r, \boldsymbol{d}_r)$ maps to a matching $m_r \in \mathcal{M}_r$. We define the intermediate state space as $\mathcal{S}_r = \mathcal{M}_r \times \Sigma_r \times \mathcal{D}_r$ and define the intermediate distribution as

$$\begin{aligned} \gamma_r(m_r, \sigma_r, \boldsymbol{d}_r|\theta) &= p(m_r|\sigma_r, \boldsymbol{d}_r) \times p(\sigma_r, \boldsymbol{d}_r|\theta) \\ &= 1\big[(\sigma_r, \boldsymbol{d}_r) \to m_r\big] \times p(\sigma_r, \boldsymbol{d}_r|\theta). \end{aligned} \tag{6.1}$$

Here, $1[(\sigma_r, \boldsymbol{d}_r) \to m_r]$ denotes the indicator function that is 1 if $(\sigma_r, \boldsymbol{d}_r)$ maps to m_r and 0 otherwise. Note that each $(\sigma_r, \boldsymbol{d}_r)$ maps to exactly one matching $m_r \in \mathcal{M}_r$ because each decision made by a knot results in it being placed in exactly one edge.

The state space that our SMC sampler operates on is defined as $\mathcal{S} = \bigcup_r \mathcal{S}_r$. We can take the proposal distribution ν^+ as the sequential decision model given in equation (4.1). This choice ensures that the state space $\mathcal{S}$ is connected starting from the initial state $s_0 = (m_0, \sigma_0, \boldsymbol{d}_{\sigma_0})$, where $m_0 = \varnothing$. It is easy to verify that the weight function reduces to 1 with this choice of the proposal for the intermediate distribution given in equation (6.1).

6.4. *Overcounting correction.* Designing an SMC sampler for a combinatorial state space requires careful attention to the possibility of an overcounting problem which may lead to biased estimates of the desired quantities [Wang, Bouchard-Côté and Doucet (2015)]. The overcounting problem arises when a certain choice of proposal results in multiple paths that can lead to the same state. Consider a graph with four partitions and one node in each partition (see Figure 4(b)). The overcounting problem for this case is illustrated in Figure 7. In this figure we can see that there are three paths leading to the state $\{\{1, 2, 3\}\}$ starting from the initial

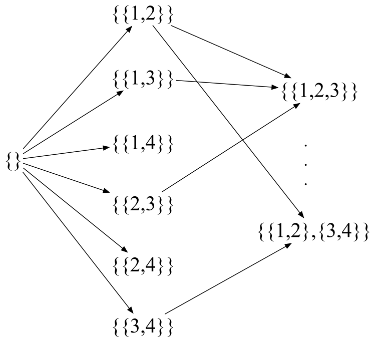


FIG. 7. *The state* $\{\{1,2\},\{3,4\}\}$ *can be reached by two distinct paths from the initial state whereas the state* $\{\{1,2,3\}\}$ *can be reached by three distinct paths.*

state, whereas there are two paths leading to the state $\{\{1,2\},\{3,4\}\}$. Approximation of any desired quantities using the SMC described in Section 6.3 would lead to bias if this overcounting problem was not corrected. A solution to this problem is to incorporate the backward kernel ν^- [Wang, Bouchard-Côté and Doucet (2015)]. One particular form of the backward kernel that works is

$$\nu^-(s' \to s) = |\mathcal{Q}(s')|^{-1} \times 1[\nu^+(s \to s') > 0], \tag{6.2}$$

where $\mathcal{Q}(s')$ is the number of possible parent states of s' and $\nu^+(s \to s') > 0$ if $s' \in \mathcal{S}$ can be proposed in one step starting from $s \in \mathcal{S}$. This leads to the weight computation step as [Del Moral, Doucet and Jasra (2006)]

$$\begin{aligned}\alpha(s_{r-1,a_r^n} \to s_{r,n}) &= \frac{\gamma_r(s_{r,n}) \times \nu^-(s_{r,n} \to s_{r-1,a_r^n})}{\gamma_{r-1}(s_{r-1,a_r^n}) \times \nu^+(s_{r-1,a_r^n} \to s_{r,n})} \\ &= \nu^-(s_{r,n} \to s_{r-1,a_r^n}).\end{aligned}$$

For the decision model corresponding to the knot matching application (see Section 4.2.2), we have compiled a list of possible cases and the number of possible parents for each of the cases:

- If $s \in \mathcal{S}$ does not contain any singleton edge:
  - For any edge with two nodes, if it contains
    1. two visited nodes, count two possible parent states.
    2. one visited node, count one possible parent state.
  - For any edge with three nodes, if it contains
    1. two visited nodes, count two possible parent states.
    2. three visited nodes, count six possible parent states.
- If there is at least one singleton edge in $s \in \mathcal{S}$:

- For any edge with one node, count one possible parent.
- For any edge with two nodes, if it contains
    1. one visited node, this state is not reachable under this model.
    2. two visited nodes, then count two possible parent states.
- For any edge with three nodes, if it contains
    1. two visited nodes, then return zero possible parent state.
    2. three visited nodes, then count three possible parent states.

Note that if a singleton set exists in a state and if there are two visited nodes in a three-matching, then undoing the move performed by one of the visited nodes in the three-matching breaks it into two-matching which produces a state where a singleton cannot have been attained. Hence, the last move must have been made by one of the singletons which means there can only be one parent state (which is accounted for by the singleton edge). Figure 8 is an illustration of different cases.

Note that the sequential decision model is used for sampling the permutation and the decisions in both the SMC sampler for matching and the SIS with resampling used in the Monte Carlo E-step for parameter estimation. And yet, one leads to an overcounting problem and the other does not. An important yet subtle distinction is in the state spaces of the two algorithms. When sampling matchings, the permutation and the decision sequences serve as auxiliary variables in the construction of matching (i.e., proposal distribution for matching). The multiple paths problem is a result of the choice of proposal that we make. In contrast as we seek to maximize the model parameters by integrating out the latent permutation and the

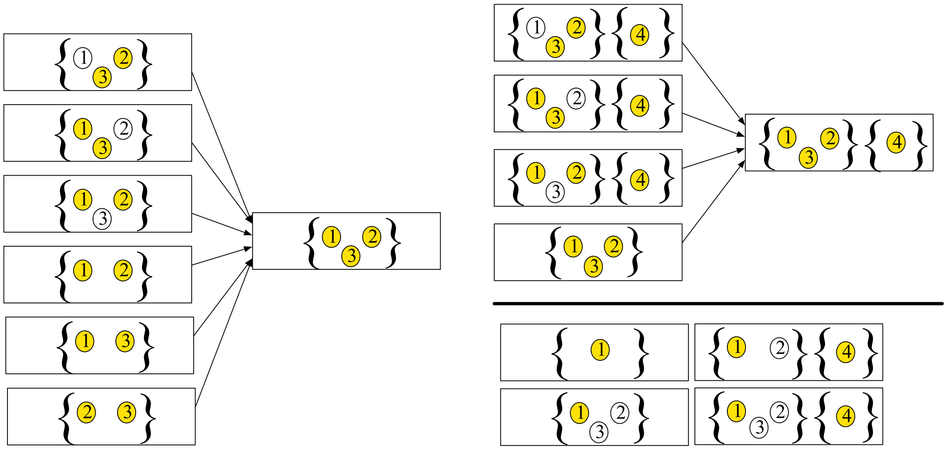


FIG. 8. *The possible parent states for different cases. Left: a state containing a three-matching where all three nodes have been visited. Top right: containing a three-matching and a singleton where all four nodes have been visited. Bottom right: example of states that are not permitted under the decision model for the knot-matching application (see Section 4.2.2).*

decisions given the observed matchings in the MC-EM procedure, the permutation and the decisions are the main variables of interest in the SIS with resampling algorithm.

6.5. *Evaluation metrics.* In this section we describe two approaches to evaluating the performance using the samples generated from the matching sampler.

6.5.1. *Single sample prediction.* We can obtain a single sample prediction by choosing the particle with the highest likelihood, denoted $\hat{m} \in \mathcal{M}$. We can then compute the prediction accuracy as

$$a(\hat{m}, m_{\text{true}}) = \frac{1}{|m_{\text{true}}|} \sum_{e \in m_{\text{true}}} 1[e \in \hat{m}], \tag{6.3}$$

where m_{true} denotes the true matching. Note that $a(\hat{m}, m_{\text{true}}) \in [0, 1]$ and it is equal to 1 when $\hat{m} = m_{\text{true}}$.

6.5.2. *Jaccard index.* To assess the entire particle population, we use Jaccard index which is the commonly used metric for computing the similarity coefficient [Levandowsky and Winter (1971)]. We can compute the Jaccard index to evaluate the deviation of each of the SMC particles from the true matching. Jaccard index is defined on two sets A and B as follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{6.4}$$

We can use Jaccard index to evaluate a particle $m_n \in \mathcal{M}$ as follows. For each node v we find the edge that contains v in m_n as well as in m_{true}. We will denote these edges by $m_n(v)$ and $m_{\text{true}}(v)$. Then, we compute the Jaccard index between the n-th particle and the truth by

$$J(m_n, m_{\text{true}}) = \frac{1}{|V|} \sum_{v \in V} J\big(m_n(v), m_{\text{true}}(v)\big). \tag{6.5}$$

Note that the minimum value for $J(m_n(v), m_{\text{true}}(v))$ is $1/5$ for the knot matching application; both $m_n(v)$ and $m_{\text{true}}(v)$ must contain v yielding the minimum numerator of 1 and the maximum denominator is attained when $|m_n(v)| = |m_{\text{true}}(v)| = 3$. The maximum value is 1 if the two edges contain the same set of nodes.

7. Generating synthetic data.

This section presents a mechanism to simulate synthetic boards that closely mimics the real data. This simulation mechanism is needed partly due to the prohibitive cost associated with obtaining the real data which limits the study of knot formulation and implementation of new covariates to accurately capture the variety of knot shapes.

We first conceptualize tree branches as approximately cone-shaped objects emanating from the center of the tree trunk [Guindos and Guaita (2013)]. As the tree is cut into construction lumber, these cones intersect with rectangular prisms representing the pieces of lumber, forming knot faces on the board's surfaces. Thus synthetic boards and knot faces with known matchings and realistic geometry can be generated by simulating locations and sizes of cones representing the tree branches, and calculating the conic sections with the four planes representing the surfaces of the board. Conic sections arising from the same branch are matched knot faces.

The board is situated in 3-D Cartesian coordinates as described in Section 2, with length 5000 units (x dimension), width 300 units (y dimension) and height 150 units (z dimension). Let n_k denote the random number of knots on the board. Based on the number of knots observed in real data, we draw $n_k \sim \text{Poisson}(\rho)$ and generate n_k branches that intersect with the board. Lumber is cut so that most branches go through the two "wide" surfaces, so we initially position a branch according to the equation of a right circular cone that opens upward from the origin,

$$\frac{x^2 + y^2}{c_0^2} = z^2,$$

where the random c_0 governs the slope of the cone and we restrict $z > 0$. It is possible for branches emanating in different directions to appear in one board, so with probability 1/2 we allow the cone to open downward by reflecting it over the plane $z = 75$. We next apply a random angle of rotation to the cone around the x and y axes to mimic the variability in the angles of tree branches. Then, the center of the cone is translated to a random (x, y) location on the board. Additional variation in the sizes of knot faces is provided by a random translation in the z direction. Intersections of the final cone and the four surfaces are computed using a numerical root-finding procedure, and ellipses are fitted to the intersections representing the knot faces.

The cone simulation is repeated for each of the n_k branches required. Since different tree branches do not intersect, we impose a condition to reject simulated cones that overlap geometrically or are otherwise too close to existing cones. Consider the 3-D line segment joining the centers of two knot faces due to the same branch. Then, the line segments corresponding to different branches cannot be too close. Specifically, the minimum distance d between the two line segments should exceed the typical diameter of a branch. Hence, a simulated cone is rejected and resampled if $d < 50$ with an existing cone. This corresponds to a real distance cutoff of about 0.6 in.

The procedure for generating a board is summarized in Algorithm 1 with the specific simulation parameters used. We use this procedure to generate the samples of synthetic boards used in the computational experiments in the following section. A sample of a simulated board is shown in Figure 9.

Algorithm 1 : Synthetic board generator

```
1:  Draw $n_k \sim \text{Pois}(25)$
2:  for $i = 1, \dots, n_k$ do
3:      Draw cone parameters: slope $c_0 \sim \text{Unif}[0.025, 0.05]$, orientation $s \sim \text{Bern}(0.5)$
4:      Draw rotation angles: $\theta_x, \theta_y \overset{iid}{\sim} \text{Unif}[-\pi/6, \pi/6]$
5:      Draw center $(x_t, y_t)$: $x_t \sim \text{Unif}[0, 5000]$, $y_t \sim \text{Unif}[0, 300]$
6:      Draw $z$ translation: $z_t \sim (2s - 1)\,\text{Unif}[0, 500]$
7:      for $j = 1, 2, 3, 4$ do
8:          if cone intersects with surface $j$ then
9:              Compute center and covariance matrix for ellipse of conic section on surface $j$
10:             $t_{ij} \leftarrow 1$
11:         else
12:             $t_{ij} \leftarrow 0$
13:         end if
14:     end for
15:     Compute line segment $L_i$ between ellipse centers on two surfaces with $t_{ij} = 1$
16:     for $b = 1, \dots, i - 1$ do
17:         $d_{i,b} \leftarrow$ minimum distance between $L_i$ and $L_b$
18:         if $d_{i,b} < 50$ then
19:             goto 3
20:         end if
21:     end for
22:     Define matching $m_i = \{j : t_{ij} = 1\}$
23: end for
```

8. Experimental results. This section presents the experimental results to demonstrate the performance of the methods proposed in this paper. We have 30 real lumber boards that were manually annotated (i.e., knots were matched manually) for evaluation purposes. As it is expensive to acquire additional data, we used the procedure described in Section 7 to simulate additional pieces of lumber to supplement the real data for analysis, in particular, to test the feasibility of deploying the methodology under the real time constraint.

8.1. *Preliminary experiments.* Before tackling the knot-matching data, we perform experiments to validate various components of the model and the methods proposed in this paper.

8.1.1. *Validation of parameter estimation procedure.* We begin with a simple parameter estimation experiments. We simulate a synthetic graph as follows:

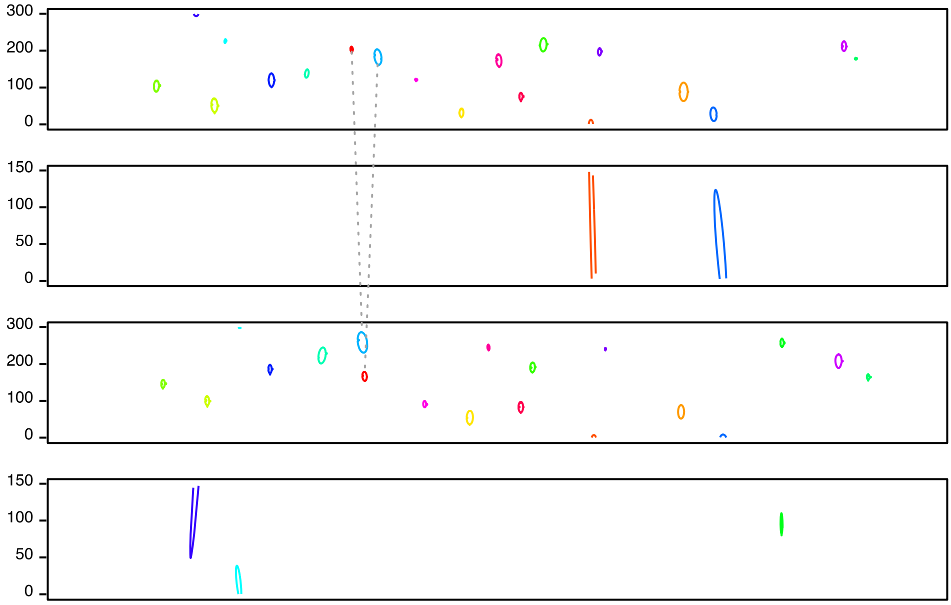
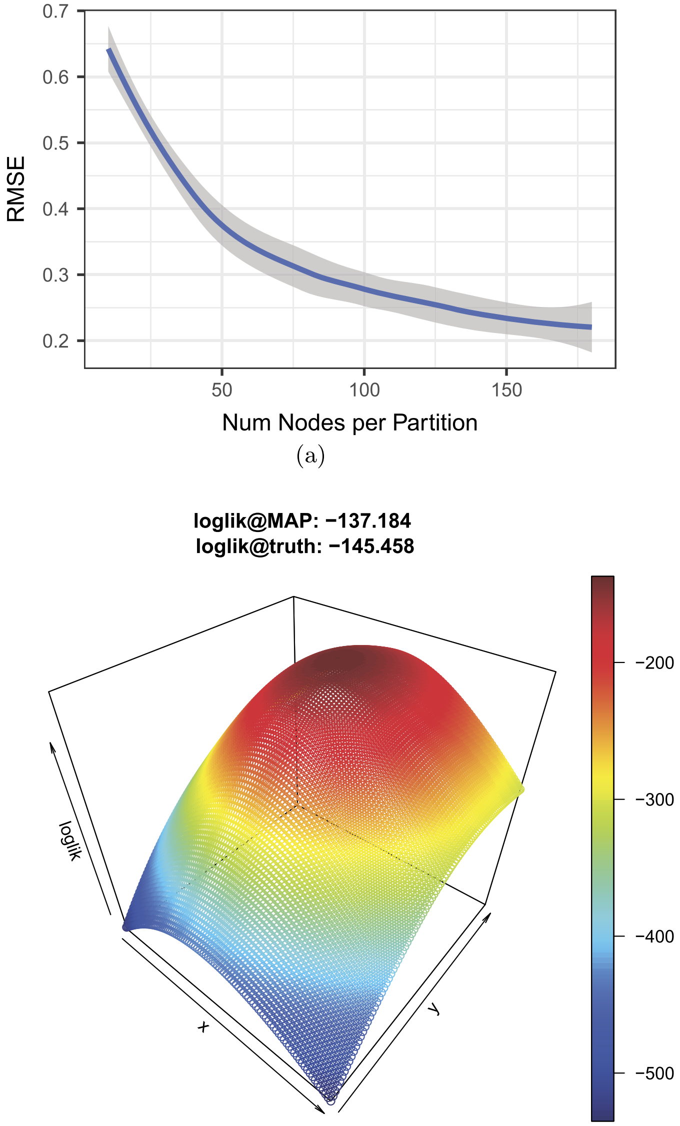


FIG. 9. *A randomly generated board and the knots from Algorithm 1. The wide surfaces range over y-axis in* $[0, 300]$ (*shown on the first and third rows*), *and the narrow surfaces range over z-axis in* $[0, 150]$ (*shown on the second and last rows*). *The horizontal axis corresponds range over* $[0, 5000]$. *The color coding identifies knot faces stemming from the same cone* (*representing a tree branch*). *The matching algorithm produced two incorrect matches for this sample which are indicated by the dotted lines.*

1. Sample the parameters, $\theta_j \sim N(0, \tau^2)$.
2. Generate N nodes per partition.
3. For each node sample the covariates $f_j \sim N(0, \zeta^2)$ for $j = 1, \dots, d$, where d denotes the number of covariates.
4. Sample σ from uniform distribution over the permutation.
5. Sample the decisions $d_\sigma \sim p(\cdot|\sigma, \theta)$ (using equation (4.1)).

We can repeat the synthetic graph generation process I times to obtain a set of labeled matchings $\{(m^i, \sigma^i, \boldsymbol{d}_{\sigma^i})\}_{i=1}^I$. Given this dataset, we carried out experiments to validate the MAP estimator of the parameters. We used the true σ that was used for generating each of the matchings. We have shown the root mean squared error, $d^{-1}\|\hat{\theta}_{\mathrm{MAP}} - \theta_{\mathrm{true}}\|_2$, when $d = 2$ in Figure 10(a). As the number of nodes in the graph is increased, the accuracy of the estimation improves as expected. We have also generated a surface plot of the posterior when $d = 2$ (two covariates) in Figure 10(b). Note that the response surface that we are optimizing over is convex and the MAP estimate attains a higher value of the log-likelihood compared to the truth. Figure 10(a) was obtained using $I = 10$. We fixed the number of partitions to *two*. The standard error estimates were obtained using LOESS in the R package



(a)

(b)

FIG. 10. *Experiments where the sequence σ is given.* (*a*) *The plot of RMSE as the number of nodes is increased.* (*b*) *The sample posterior surface showing that the MAP estimate correctly finds the mode of the posterior.*

ggplot2 [Wickham (2009)]. Figure 10(b) was obtained by evaluating the likelihood function over a grid of parameter values. This experiment serves to verify the correctness of our parameter estimation procedure.

$\{\{1,2,3\},\{4\}\}$

$\{\{1,2,4\},\{3\}\}$

$\{\{1,3,4\},\{2\}\}$

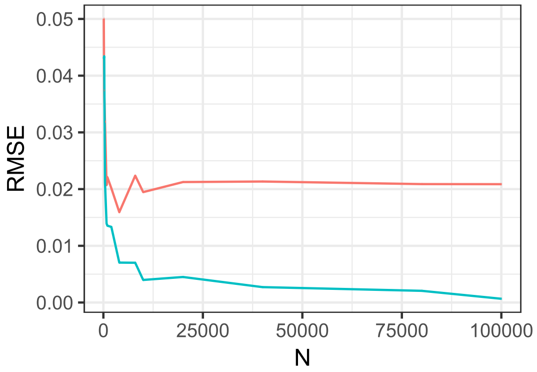
$\{\{2,3,4\},\{1\}\}$

$\{\{1,2\},\{3,4\}\}$

$\{\{1,3\},\{2,4\}\}$

$\{\{1,4\},\{2,3\}\}$

(a)



(b)

FIG. 11. *Overcounting problem illustrated on sampling from uniform graph matching.* (*a*) *The number of possible* $\{2, 3\}$*-matchings for a graph with four partitions and one node in each partition.* (*b*) *The root mean squared error with overcounting correction* (*blue*), *without the overcounting correction* (*red*).

8.1.2. *Overcounting correction experiments.* This subsection illustrates the overcounting problem, and why it needs to be addressed to sample graph matching using SMC. To that end, we assume a scenario where we want to sample graph matching from the uniform distribution over all possible configurations permitted by our choice of the decision model (e.g., the decision model for knot matching given in Section 4.2.2). For illustrative purposes suppose we have a simple example of a quadripartite graph with one node in each partition. Note that this decision model is restricted to $\{2, 3\}$-matchings, so there are total of seven matchings possible for this graph and, hence, we expect the probability of sampling a matching to be $1/7$ (see Figure 11(a)).

We have computed the estimate of the probability of each matching configuration from the SMC population, $\hat{p}_m$, and computed the root mean squared error, $\sqrt{7^{-1}\sum_{m\in\mathcal{M}}(\hat{p}_m - 1/7)^2}$. In Figure 11(b) we show that the RMSE tends to 0 as the number of particles used in the SMC is increased (the blue curve). On the other hand the RMSE stabilizes around 0.02 when the overcounting problem is ignored (the red curve).

8.2. *Data analysis.* In this section we analyze the simulated and real lumber data.

8.2.1. *Real data analysis.* In this section, we evaluate our methods on the 30 boards that had been manually annotated. First, we illustrate the parameter estimation procedure that was carried out using an MC-EM procedure. The sample size used for the E-step is kept at 100 for the first 10 iterations of MC-EM which is increased to 500 onward to reduce the Monte Carlo error across the iterations. The

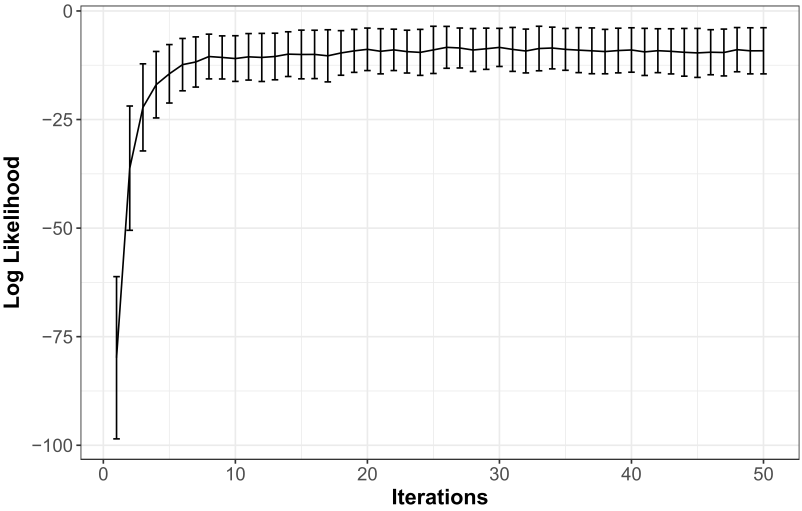


FIG. 12. *The plot of the $\tilde{Q}$ function versus iterations. The error bars correspond to $\tilde{Q}$ plus/minus two times the sample standard deviation* (*i.e.*, $\tilde{Q} \pm 2\hat{\sigma}$). *The convergence of MC-EM seems to have been reached in* 10 *iterations.*

convergence of MC-EM is monitored by plotting $\tilde{Q}$ across the iterations. In Figure 12 we show a plot of the $\tilde{Q}$ function across MC-EM iterations with the error bars computed using the standard deviation of the Monte Carlo samples to $\tilde{Q}$ at each iteration of MC-EM. The figure suggests that convergence is reached in about 10 iterations. The trajectory of the parameters across MC-EM iterations is shown in Figure 13. This plot shows that distance based covariates play important roles compared to area based covariates. The value for λ is set to 1 for the experiments.

To evaluate predictive performance, we perform leave-one-out cross validation. That is, we leave one board out from the MC-EM inference procedure (i.e., obtain MAP estimate using the remaining 29 boards). Then, we sample graph matchings on the held-out board to be evaluated using single sample prediction accuracy and the Jaccard index described in Section 6.5. With $\lambda = 1$, the overall accuracy is $375/384 = 0.977$ using a single sample prediction. The board-by-board performance is shown in Figure 14. We have experimented with $\lambda = 0.1$ and $\lambda = 10$ as well and found the single sample prediction performance to be comparable to $\lambda = 1$.

8.2.2. *Simulated data.* A data simulation procedure is helpful for future research in the automatic strength grading of lumber as it can be used to calibrate the performance of matching methodology presented in the paper. In particular there are various types of knots that we have not been able to model due to the limitation in the dataset. The simulation provides a testbed to develop new models for knots and new features for knot matching for the application experts.

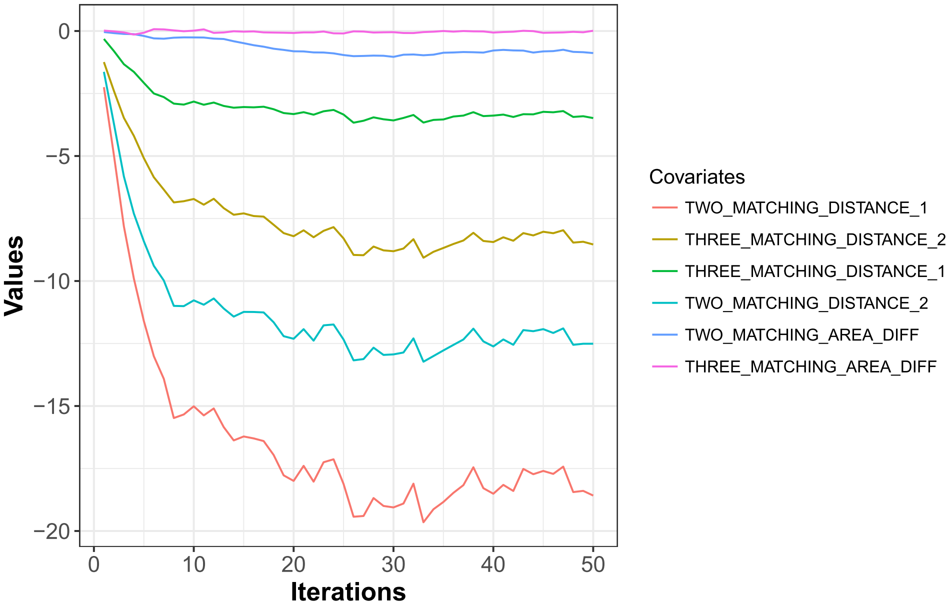


FIG. 13. *The trajectory of parameters in the MC-EM training versus iterations. The distance based covariates seem to play important roles in determining the correct matches compared to the area based covariates.*

Another way to use the simulated data is to test the feasibility of deploying the SMC sampler in real time. As the end goal is to deploy the matching mechanism developed here to mills that operate under real time constraints, it is important to study the time it takes to generate samples using SMC. To that end, we simulated 100 boards. To speed up the sampling procedure, we segmented each board into multiple subgraphs. This segmentation procedure was carried out based on distance so that knots within certain distances are placed into the same subgraph. The SMC sampler was executed locally within each subgraph, and this helped to significantly reduce the time to draw samples since there are less decisions to consider at each iteration of an SMC.

We have plotted the timing results in Figure 15. The figure depicts the scatter plot of the timing results for 100 simulated boards as well as 30 real boards when the number of particles is set to 1000 against the number of knot faces on the board. Observe that for most boards, it only takes a fraction of a second for sampling to complete. For completeness, we carried out a two-fold validation procedure to quantify the performance of our methodologies on the simulated dataset. We attained the single sample prediction accuracy of 93% on the 100 simulated boards. We note that an error rate of 7% is reasonable when the sizes and locations of knots are generated from uniform distributions. For example, we have identified two incorrect matches on the simulated board shown in Figure 9. In this figure the

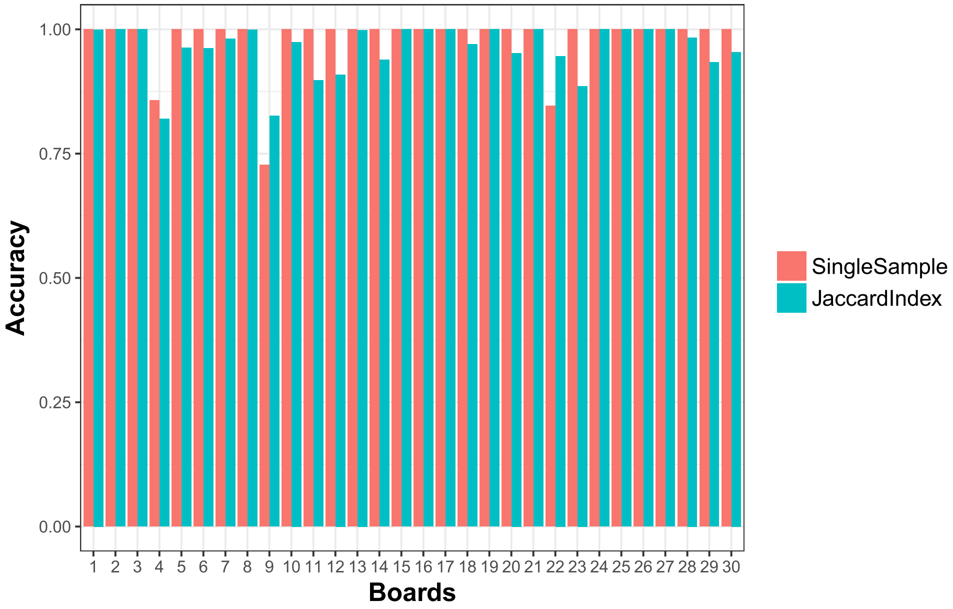


FIG. 14. *Single sample prediction accuracy and Jaccard index evaluation of the matching samples generated by SMC computed on the real data for each board. The single sample prediction accuracy is perfect for all but three boards. The quality of the matchings generated for each board by SMC appears reasonable based on the values of Jaccard index.*

blue knot face on surface 1 is incorrectly matched to the red knot face on surface 3. This is due to the small values in the size and distance covariates between those two knot faces, and so the erroneous matching has a higher log-likelihood compared to the truth.

9. Conclusion and discussion. We developed methods for the novel knot-matching application which can be formulated as a four-partite hypergraph matching problem. This is an important step toward automating the grading of lumber, one where statistical inferential methods can be used to produce not just a single strength prediction value but a posterior predictive interval that captures any uncertainties encountered in the process. Furthermore, the model could in principle admit a full Bayesian approach via particle MCMC methodology [Andrieu, Doucet and Holenstein (2010)]. In this paper, the covariates we used encode only basic characteristics of the knots, based on distance, size and surface information. Nonetheless, a high level of performance was achieved with these simple covariates. The framework we have laid out in this paper is general and allows users to craft and incorporate specialized covariates to further improve matching accuracy. For example, additional information that might be incorporated include the rotation angle of the knot faces and/or more detailed shape information on the knots.

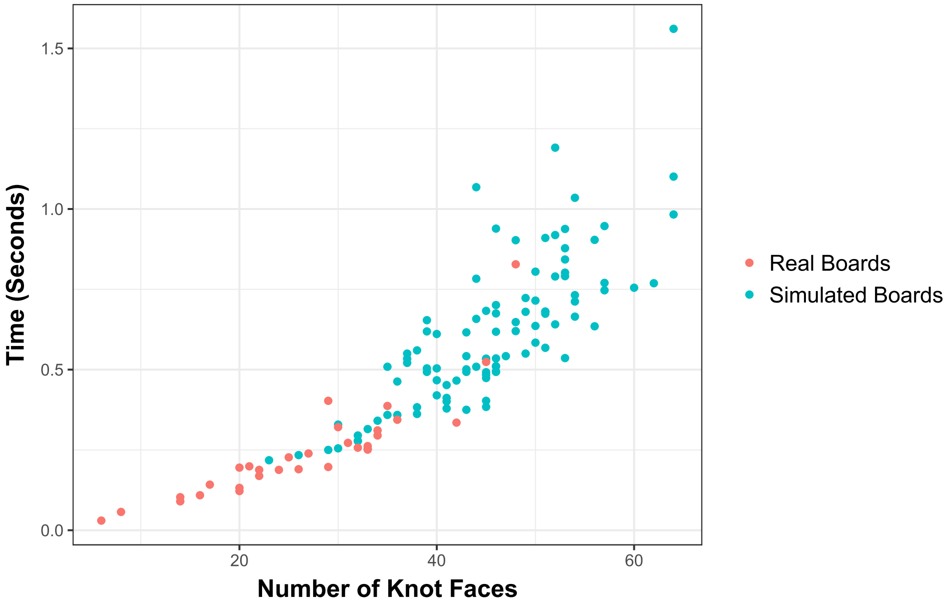


FIG. 15. *Timing results on the simulated and the real boards. The prediction times are within a second for the real boards. The timing results on the simulated boards serve to test the feasibility of deploying the SMC sampler in lumber mills.*

The sequential decision model and an SMC sampler for sampling of matching was initially developed in our previous work [Jun et al. (2017)]. However, the previous work had its limitations as it focused on the standard K-partite graphs and the supervised learning algorithm was not completely developed. In this paper we completed the development of inference algorithm for supervised setting using Monte Carlo expectation maximization algorithm Wei and Tanner (1990) and a solution to the overcounting problem for knot matching problem that includes edges of size 3 which allows us to completely solve the application at hand. One of the main intentions was to clearly convey the overcounting problem for combinatorial spaces to the readers and demonstrate how one can develop a consistent SMC algorithm.

Future work remains to complete an automatic lumber strength grading pipeline. We shall develop an enhanced statistical model for strength prediction, using the output from our knot matching methodology as an input for producing the strength estimate. With accurate knot matchings and uncertainty appropriately quantified, we anticipate that our contributions will have practical impact.

Acknowledgments. The authors thank FPInnovations, in particular, Zarin Pirouz, Bruce Lehmann, and Alex Precosky for collection and processing of the data.

REFERENCES

ANDRIEU, C., DOUCET, A. and HOLENSTEIN, R. (2010). Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **72** 269–342. MR2758115

BERG-KIRKPATRICK, T., BOUCHARD-CÔTÉ, A., DENERO, J. and KLEIN, D. (2010). Painless unsupervised learning with features. In *Proceedings of the North American Chapter of the Association for Computational Linguistics* (*NAACL*10) **8** 582–590.

BONDY, J. A. and MURTY, U. S. R. (1976). *Graph Theory with Applications*. Elsevier, New York. MR0411988

BOUCHARD-CÔTÉ, A. and JORDAN, M. I. (2010). Variational inference over combinatorial spaces. In *Advances in Neural Information Processing Systems* 23 (*NIPS*) **23** 280–288.

BOUCHARD-CÔTÉ, A., SANKARARAMAN, S. and JORDAN, M. I. (2012). Phylogenetic inference via sequential Monte Carlo. *Syst. Biol.* **61** 579–593. DOI:10.1093/sysbio/syr131.

BRÄNNSTRÖM, M. (2009). The impact of a strength grading process on sawmill profitability and product quality. *BioResources* **4** 1430–1454.

CAETANO, T. S., MCAULEY, J. J., CHENG, L., LE, Q. V. and SMOLA, A. J. (2009). Learning graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **31** 1048–1058.

CAO, Z., QIN, T., LIU, T., TSAI, M. and LI, H. (2007). Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the* 24*th International Conference on Machine Learning* 129–136. ACM, New York.

CARON, F., TEH, Y. W. and MURPHY, T. B. (2014). Bayesian nonparametric Plackett–Luce models for the analysis of preferences for college degree programmes. *Ann. Appl. Stat.* **8** 1145–1181. MR3262549

CASTÉRA, P., FAYE, C. and EL OUADRANI, A. (1996). Prevision of the bending strength of timber with a multivariate statistical approach. In *Annales des Sciences Forestières* **53** 885–896. EDP Sciences.

DAVAL, V., POT, G., BELKACEMI, M., MERIAUDEAU, F. and COLLET, R. (2015). Automatic measurement of wood fiber orientation and knot detection using an optical system based on heating conduction. *Opt. Express* **23** 33529–33539.

DEL MORAL, P., DOUCET, A. and JASRA, A. (2006). Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **68** 411–436. MR2278333

DEMPSTER, A. P., LAIRD, N. M. and RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B* **39** 1–38. MR0501537

DOUCET, A. and JOHANSEN, A. M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In *The Oxford Handbook of Nonlinear Filtering* 656–704. Oxford Univ. Press, Oxford. MR2884612

GREEN, D. W., ROSS, R. J. and MCDONALD, K. A. (1994). Production of hardwood machine stress rated lumber. In *Proceedings of* 9*th International Symposium on Nondestructive Testing of Wood* 141–150.

GUINDOS, P. and GUAITA, M. (2013). A three-dimensional wood material model to simulate the behavior of wood with any type of knot at the macro-scale. *Wood Sci. Technol.* **47** 585–599.

HANSEN, B. B. (2004). Full matching in an observational study of coaching for the SAT. *J. Amer. Statist. Assoc.* **99** 609–618. MR2086387

HIETANIEMI, R., HANNUKSELA, J. and SILVEÉN, O. (2011). Camera based lumber strength classification system. In *MVA*2011 *IAPR Conference on Machine Vision Applications* 251–254.

HIETANIEMI, R., LÓPEZ, M. B., HANNUKSELA, J. and SILVÉN, O. (2014). A real-time imaging system for lumber strength prediction. *For. Prod. J.* **64** 126–133.

HOLMES, I. and RUBIN, G. M. (2001). Pairwise RNA structure comparison with stochastic context-free grammars. In *Proceedings of the Pac. Symp. Biocomputing* **7** 163–174.

JUN, S.-H., WONG, S. W., ZIDEK, J. and BOUCHARD-COTE, A. (2017). Sequential graph matching with sequential Monte Carlo. In *Artificial Intelligence and Statistics* 1075–1084.

KUHN, H. W. (1955). The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2** 83–97. MR0075510

LEVANDOWSKY, M. and WINTER, D. (1971). Distance between sets. *Nature* **234** 34–35.

LIU, D. C. and NOCEDAL, J. (1989). On the limited memory BFGS method for large scale optimization. *Math. Program.* **45** 503–528. MR1038245

LU, B. and ROSENBAUM, P. R. (2004). Optimal pair matching with two control groups. *J. Comput. Graph. Statist.* **13** 422–434. MR2063993

LUNTER, G., DRUMMOND, A. J., MIKLÓS, I. and HEIN, J. (2005). Statistical alignment: Recent progress, new applications, and challenges. In *Statistical Methods in Molecular Evolution*. *Stat. Biol. Health* 375–405. Springer, New York. MR2161838

MØLLER, J., PETTITT, A. N., REEVES, R. and BERTHELSEN, K. K. (2006). An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika* **93** 451–458. MR2278096

OLSSON, A., OSCARSSON, J., SERRANO, E., KÄLLSNER, B., JOHANSSON, M. and ENQUIST, B. (2013). Prediction of timber bending strength and in-member cross-sectional stiffness variation on the basis of local wood fibre orientation. *Eur. J. Wood Prod.* **71** 319–333.

PAPADIMITRIOU, C. H. and STEIGLITZ, K. (1982). *Combinatorial Optimization*: *Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ. MR0663728

PETTERSON, J., YU, J., MCAULEY, J. J. and CAETANO, T. S. (2009). Exponential family graph matching and ranking. In *Advances in Neural Information Processing Systems* 1455–1463.

PLACKETT, R. L. (1975). The analysis of permutations. *J. R. Stat. Soc. Ser. C. Appl. Stat.* **24** 193–202. MR0391338

WANG, L., BOUCHARD-CÔTÉ, A. and DOUCET, A. (2015). Bayesian phylogenetic inference using a combinatorial sequential Monte Carlo method. *J. Amer. Statist. Assoc.* **110** 1362–1374. MR3449032

WEI, G. C. and TANNER, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J. Amer. Statist. Assoc.* **85** 699–704.

WICKHAM, H. (2009). *Ggplot2*: *Elegant Graphics for Data Analysis*. Springer, New York.

WONG, S. W. K., LUM, C., WU, L. and ZIDEK, J. V. (2016). Quantifying uncertainty in lumber grading and strength prediction: A Bayesian approach. *Technometrics* **58** 236–243. MR3488302

S.-H. JUN
J. V. ZIDEK
A. BOUCHARD-CÔTÉ
DEPARTMENT OF STATISTICS
UNIVERSITY OF BRITISH COLUMBIA
2329 WEST MALL
VANCOUVER, BRITISH COLUMBIA
CANADA V6T 1Z4
E-MAIL: s2jun.uw@gmail.com
jim@stat.ubc.ca
bouchard@stat.ubc.ca

S. W. K. WONG
DEPARTMENT OF STATISTICS
AND ACTUARIAL SCIENCE
UNIVERSITY OF WATERLOO
200 UNIVERSITY AVENUE WEST
WATERLOO, ONTARIO
CANADA N2L 3G1
E-MAIL: samuel.wong@uwaterloo.ca