Sequential Monte Carlo methods for graphical models

Christian A. Naesseth
Linköping University, 581 83 Linköping, Sweden

Fredrik Lindsten
Linköping University, 581 83 Linköping, Sweden

Thomas B. Schön
Uppsala University, 751 05 Uppsala, Sweden

Abstract

Inference in probabilistic graphical models (PGMs) does typically not allow for analytical solutions, confining us to various approximative methods. We propose a sequential Monte Carlo (SMC) algorithm for inference in general PGMs. Via a sequential decomposition of the PGM we find a sequence of auxiliary distributions defined on a monotonically increasing sequence of probability spaces. By targeting these auxiliary distributions using purpose built SMC samplers we are able to approximate the full joint distribution defined by the graphical model. Our SMC sampler also provides an unbiased estimate of the partition function (normalization constant) and we show how it can be used within a particle Markov chain Monte Carlo framework. This allows for better approximations of the marginals and for unknown parameters to be estimated. The proposed inference algorithms can deal with an arbitrary graph structure and the domain of the random variables in the graph can be discrete or continuous.

1. Introduction

Bayesian inference in statistical models involving a large number of latent random variables is in general a difficult problem. This renders inference methods that are capable of efficiently utilizing structure important tools. Probabilistic Graphical Models (PGMs) are an intuitive and useful way to represent and make use of underlying structure in probability distributions. These types of models have a wide range of applications in many areas, including, but is not limited to, image analysis (Li, 2009), information theory (Molkaraie & Loeliger, 2013), social networks (Everitt, 2012), statistical physics and many other (Jordan, 2004). Most current methods to reason about PGMs are either variational methods (Wainwright & Jordan, 2008) or Markov chain Monte Carlo (MCMC) methods (Robert & Casella, 2004).

Our main contribution is a sequential decomposition of PGMs allowing us to derive a new class of sequential Monte Carlo (SMC) algorithms for these models. The proposed algorithms operate on factor graphs, which means that they are applicable to both directed and undirected graphical models. Furthermore, we do not have to make any assumptions concerning the structure of the graph and the variables can be continuous or discrete valued.

From the sequential decomposition we define a sequence of intermediate (auxiliary) target distributions defined on a monotonically increasing sequence of probability spaces. This sequence of distributions is then targeted by an SMC sampler, enabling us approximate the full joint distribution defined by the PGM. As a byproduct, the SMC algorithm provides an unbiased estimate of the partition function (normalization constant) of the model. This is a very interesting quantity in many applications (see Section 3.3 for examples), but it is also known to be very difficult to estimate using standard methods, such as MCMC (Hamze & de Freitas, 2005). We also make use of the proposed SMC algorithm to design efficient, high-dimensional MCMC kernels for the latent variables of the PGM in a particle MCMC framework. This enables inference about the latent variables as well as learning of unknown model parameters in an MCMC setting.

During the last decade there has been a lot of work on how to leverage SMC algorithms (Doucet et al., 2001) to solve inference problems in PGMs. The first of these algorithms

where the so called PAMPAS approach by Isard (2003) and nonparametric belief propagation by Sudderth et al. (2003; 2010). They both rely on a particle approximation of the messages sent between variables in a message passing algorithm. For loopy graphs, these methods are thus approximate even in the limit of the number of Monte Carlo samples. Since then, several different variants and refinements have been proposed by e.g. Briers et al. (2005); Ihler & Mcallester (2009); Frank et al. (2009). Another branch of SMC methods for graphical models that do not rely on message passing algorithms have been suggested by Hamze & de Freitas (2005). This method builds on the SMC sampler by Del Moral et al. (2006), where the initial target is a spanning tree of the original graph. At subsequent iterations the target is increased by adding edge by edge until all the edges of the graph have been added. Everitt (2012) extends these ideas to learn parameters using particle MCMC (Andrieu et al., 2010). Yet another take is provided by Carbonetto & de Freitas (2007), where an SMC sampler is combined with mean field approximations. Our method differs from these methods in that we use a sequential addition of edges and nodes, instead of a sequential addition of edges alone.

In Section 2 we give a brief introduction to probabilistic graphical models. We present our main contributions in Sections 3 and 4, where we present the proposed SMC and particle MCMC algorithms, respectively. In Section 5 we illustrate the algorithms in numerical examples and, finally, in Section 6 we draw conclusions.

2. Graphical models

A graphical model is a probabilistic model which factorizes according to the structure of an underlying graph \( \mathcal{G} = \{V, \mathcal{E}\} \), with vertex set \( V \) and edge set \( \mathcal{E} \). By this we mean that the joint probability density function (PDF) of the set of random variables indexed by \( V \), \( X_V := \{x_1, \ldots, x_{|V|}\} \), can be represented as a product of factors over the cliques of the graph:

\[
p(X_V) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(X_C),
\]

where \( \mathcal{C} \) is the set of cliques in \( \mathcal{G} \), \( \psi_C \) is the factor for clique \( C \), and \( Z = \int \prod_{C \in \mathcal{C}} \psi_C(x_C) \, dX_V \) is the partition function.

We will frequently use the notation \( X_I = \bigcup_{i \in I} \{x_i\} \) for some subset \( I \subseteq \{1, \ldots, |V|\} \) and we write \( X_I \) for the range of \( X_I \) (i.e., \( X_I \in X_I \)). To make the interactions between the random variables explicit we define a factor graph \( \mathcal{F} = \{V, \Psi, \mathcal{E}'\} \) corresponding to \( \mathcal{G} \). The factor graph consists of two types of vertices, the original set of random variables \( X_V \) and the factors \( \Psi = \{\psi_C : C \in \mathcal{C}\} \). The edge set \( \mathcal{E}' \) consists only of edges from variables to factors. In Figure 1(a) we show a simple example of an undirected graphical model, and the corresponding factor graph is shown in Figure 1(b).

![Figure 1. Example of an undirected graphical model and a corresponding factor graph.](image)

Both directed and undirected graphs can be represented by factor graphs.

3. Sequential Monte Carlo

In this section we propose a way to sequentially decompose a graphical model which we then make use of to design an SMC algorithm for the PGM.

3.1. Sequential decomposition of graphical models

SMC methods can be used to approximate a sequence of probability distributions on a sequence of probability spaces of increasing dimension. This is done by updating recursively a set of samples—or particles—with corresponding nonnegative importance weights. The typical scenario is that of state inference in state-space models, where the probability distributions targeted by the SMC sampler are the joint smoothing distributions of a sequence of latent states conditionally on a sequence of observations; see e.g., Doucet & Johansen (2011) for applications of this type. However, SMC is not limited to these cases and it is applicable to a much wider class of models.

To be able to use SMC for inference in PGMs we have to define a sequence of target distributions for the sampler. However, these target distributions do not have to be marginal distributions under \( p(X_V) \). Indeed, as long as the sequence of target distributions is constructed in such a way that, at some final iteration, we recover \( p(X_V) \), all the intermediate target distributions may be chosen quite arbitrarily.

This is key to our development, since it lets us use the structure of the PGM to define a sequence of intermediate target distributions for the sampler. We do this by a so called sequential decomposition of the graphical model. This amounts to simply adding factors to the target distribution, from the product of factors in (1), at each step of the algorithm and iterate until all the factors have been added. Constructing an artificial sequence of intermediate target distributions for an SMC sampler is a simple, albeit under-
Sequential Monte Carlo methods for graphical models

utilized, idea as it opens up for using SMC samplers for inference in a wide range of probabilistic models; see e.g., Bouchard-Côté et al. (2012); Del Moral et al. (2006) for a few applications of this approach.

Given a graph $\mathcal{G}$ with cliques $\mathcal{C}$, let $\{\psi_k\}_{k=1}^K$ be a sequence of factors defined as follows:

$$
\psi_k(X_{I_k}) = \prod_{C \in \mathcal{C}_k} \psi_C(X_C),
$$

where $\mathcal{C}_k \subseteq \mathcal{C}$ are chosen such that $\bigcup_{k=1}^K \mathcal{C}_k = \mathcal{C}$ and $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$, $i \neq j$, and where $I_k \subseteq \{1, \ldots, |\mathcal{V}|\}$ is the index set of the variables in the domain of $\psi_k$.

$$
I_k = \bigcup_{C \in \mathcal{C}_k} C.
$$

We emphasize that the cliques in $\mathcal{C}$ need not be maximal.

It follows that the PDF in (1) can be written as

$$
p(X_V) = \frac{1}{Z} \prod_{k=1}^K \psi_k(X_{I_k}).
$$

Principally, the choices and the ordering of the $\mathcal{C}_k$’s is arbitrary, but in practice it will affect the performance of the proposed sampler. However, in many common PGMs an intuitive ordering can be deduced from the structure of the model. We return to this in Sections 5 and 6.

The sequential decomposition of the PGM is then based on the following auxiliary quantities, defined for $k \in \{1, \ldots, K\}$ as

$$
\tilde{\gamma}_k(X_{\mathcal{L}_k}) := \prod_{i=1}^k \psi_i(X_{I_i}),
$$

where $\mathcal{L}_k := \bigcup_{i=1}^k I_i$.

By construction, $\mathcal{L}_K = \mathcal{V}$ and the joint PDF $p(X_{\mathcal{L}_K})$ will be proportional to $\tilde{\gamma}_K(X_{\mathcal{L}_K})$. Consequently, by using (2) as the basis for the target sequence for an SMC sampler, we will obtain the correct target distribution at iteration $K$.

However, a further requirement for this to be possible is that all the functions in the sequence are normalizable. For many graphical models this is indeed the case, and then we can use (2) directly as our sequence of intermediate target densities. If, however, $\int \tilde{\gamma}_k(X_{\mathcal{L}_k})dX_{\mathcal{L}_k} = \infty$ for some $k < K$, an easy remedy is to modify the target density to ensure normalizability. This is done by setting $\gamma_k(X_{\mathcal{L}_k}) = \tilde{\gamma}_k(X_{\mathcal{L}_k})q_k(X_{\mathcal{L}_k})$, where $q_k(X_{\mathcal{L}_k})$ is chosen so that

$$
\int \gamma_k(X_{\mathcal{L}_k})dX_{\mathcal{L}_k} < \infty.
$$

We set $q_k(X_{\mathcal{L}_k}) \equiv 1$ to make sure that $\gamma_k(X_{\mathcal{L}_k}) \propto p(X_{\mathcal{L}_k})$. Note that the integral in (3) need not be computed explicitly, as long as it can be established that it is finite.

With this modification we obtain, using the auxiliary quantities in (2), a sequence of unnormalized intermediate target densities for the SMC sampler as

$$
\gamma_1(X_{\mathcal{L}_1}) = q_1(X_{\mathcal{L}_1})\psi_1(X_{\mathcal{L}_1}).
$$

and

$$
\gamma_k(X_{\mathcal{L}_k}) = \gamma_{k-1}(X_{\mathcal{L}_{k-1}}) \frac{q_k(X_{\mathcal{L}_k})}{q_{k-1}(X_{\mathcal{L}_{k-1}})} \psi_k(X_{I_k}),
$$

for $k = 2, \ldots, K$. The corresponding normalized PDFs are given by

$$
\tilde{\gamma}_k(X_{\mathcal{L}_k}) = \frac{\gamma_k(X_{\mathcal{L}_k})}{Z_k},
$$

where $Z_k = \int \gamma_k(X_{\mathcal{L}_k})dX_{\mathcal{L}_k}$.

Figure 2 shows the resulting subgraphs when applying the decomposition to the factor graph example in Figure 1.

3.2. Sequential Monte Carlo for PGMs

At iteration $k$, the SMC sampler approximates the target distribution $\tilde{\gamma}_k$ by a collection of weighted particles $\{X_{\mathcal{L}_k}, w_k^i\}_{i=1}^N$. These samples define an empirical point-mass approximation of the target distribution,

$$
\hat{\gamma}_k(dX_{\mathcal{L}_k}) := \sum_{i=1}^N w_k^i \delta_{X_{\mathcal{L}_k}^i}(dX_{\mathcal{L}_k}),
$$

where $\delta_X$ is a Dirac measure at $X$. In what follows, we shall use the notation $\xi_k := X_{\mathcal{L}_k \setminus \mathcal{L}_{k-1}}$ to refer to the collection of random variables that are in the domain of $\gamma_k$, but not in the domain of $\gamma_{k-1}$. This corresponds to the collection of random variables, with which the particles are augmented at each iteration.

Initially, $\tilde{\gamma}_1$ is approximated by importance sampling. That is, we simulate independently $\{X_{\mathcal{L}_1}^i\}_{i=1}^N$ from a proposal distribution: $X_{\mathcal{L}_1}^i \sim \psi_1(\cdot)$. The samples are then assigned

![Figure 2. Sequential decomposition of the factor graph in Figure 1(b).](image-url)
importance weights according to \( w_i^t = W_1(X_{L_i}^t) \), where \( W_1(X_{L_i}) = \frac{\gamma_i(X_{L_i})}{r_i(X_{L_i})} \).

We proceed inductively and assume that we have at hand a weighted sample \( \{X_{L_i}^t, w_i^t\}_{i=1}^N \), approximating \( \gamma_k(X_{L_k}) \). This sample is propagated forward by simulating, conditionally independently given the particle generation up to iteration \( k - 1 \), as follows: For each \( i \in \{1, \ldots, N\} \), we draw an ancestor index \( a_i^k \) with probability

\[
\mathbb{P}(a_i^k = j) = \frac{\nu_{k-1}^j w_{k-1}^j}{\sum_i \nu_{k-1}^j w_{k-1}^j}, \quad j \in \{1, \ldots, N\},
\]


(6)

where \( \nu_{k-1}^j = \nu_{k-1}(X_{L_i}^{k-1}) \)—known as adjustment multiplier weights—are used in the auxiliary SMC framework to adapt the resampling procedure to the current target density \( \bar{\gamma}_k \) (Pitt & Shephard, 1999). Generating the ancestor indices corresponds to a selection—or resampling—process that will put emphasis on the most likely particles. Specifically, \( a_i^k \) is the index of the particle at the previous iteration that will be used to construct \( X_{L_i}^k \).

Given the ancestor indices, we simulate particle increments \( \{\xi_i^k\}_{i=1}^N \) from a proposal density \( r_k \) on \( X_{L_i} \setminus L_{k-1} \):

\[
\xi_i^k \sim r_k (\cdot | X_{L_i}^{a_i^k})
\]

and augment the particles as,

\[
X_{L_i}^k := X_{L_i}^{a_i^k} \cup \{\xi_i^k\}.
\]

After having performed this procedure for the \( N \) ancestor indices and particles, they are assigned importance weights \( w_i^k = W_k(X_{L_i}^k) \). The weight function, for \( k \geq 2 \), is given by

\[
W_k(X_{L_i}) = \frac{\gamma_k(X_{L_i})}{\gamma_{k-1}(X_{L_i}) \nu_{k-1}(X_{L_i}) r_k(\xi_k | X_{L_k}^{a_i^k})},
\]


(9)

where, again, we write \( \xi_k = X_{L_k} \setminus L_{k-1} \). We give a summary of the SMC method in Algorithm 1.

In the case that \( L_k \setminus L_{k-1} = \emptyset \) for some \( k \), steps (6) and (7) are superfluous. The easiest way to handle this is to simply skip these steps and directly compute importance weights. However, the previous weights \( \{w_i^{k-1}\}_{i=1}^N \) should then be retained: \( w_i^k = w_i^{k-1} \times W_k(X_{L_i}^k) \). An alternative approach is to bridge the two target distributions \( \bar{\gamma}_{k-1} \) and \( \bar{\gamma}_k \) similarly to Del Moral et al. (2006).

Since the proposed sampler for PGMs falls within a general SMC framework, standard convergence analysis applies. See e.g., Del Moral (2004) for a comprehensive collection of theoretical results on consistency, central limit theorems, and non-asymptotic bounds for SMC samplers.

**Algorithm 1 Sequential Monte Carlo**

**Perform each step for** \( i = 1, \ldots, N \).

Sample \( X_{L_i}^1 \sim r_1(\cdot) \).

Set \( w_i^1 = W_1(X_{L_i}^1) \).

**for** \( k = 2 \) to \( K \) **do**

Sample \( a_i^k \) according to (6).

Sample \( \xi_i^k \sim r_k(\cdot | X_{L_i}^{a_i^k}) \) and set \( X_{L_i}^k = X_{L_i}^{a_i^k} \cup \{\xi_i^k\} \).

Set \( w_i^k = W_k(X_{L_i}^k) \).

end for

The choices of proposal density and adjustment multipliers can quite significantly affect the performance of the sampler. The proposal density that minimizes the variance of the unnormalized importance weights is given by (Doucet & Johansen, 2011)

\[
r_k(\xi_k | X_{L_{k-1}}) = \frac{\gamma_k(X_{L_k})}{\gamma_k(X_{L_{k-1}})} \int \frac{\gamma_k(X_{L_{k-1}})}{\gamma_{k-1}(X_{L_{k-1}})} d\xi_k.
\]

If, additionally, the adjustment multipliers are chosen as

\[
\nu_{k-1}(X_{L_{k-1}}) = \int \frac{\gamma_k(X_{L_{k-1}})}{\gamma_{k-1}(X_{L_{k-1}})} d\xi_k,
\]

it follows from (9) that \( W_k(X_{L_k}) \equiv 1 \). In this case, the SMC sampler is said to be *fully adapted*.

### 3.3. Estimating the partition function

The partition function of a graphical model is a very interesting quantity in many applications. Examples include statistical mechanics where it is related to the free energy of a system of objects, information theory where it is related to the capacity of a channel, and likelihood-based learning of the parameters of the PGM. However, as stated by Hamze & de Freitas (2005), estimating the partition function of a loopy graphical model is a “notoriously difficult” task. Indeed, even for discrete problems simple and accurate estimators have proved to be elusive, and MCMC methods do not provide any simple way of computing the partition function.

On the contrary, SMC provides a straightforward estimator of the normalizing constant (i.e. the partition function), given as a byproduct of the sampling method according to

\[
\hat{Z}_k^N := \left( \frac{1}{N} \sum_{i=1}^N w_i^k \right) \left\{ \prod_{t=1}^{k-1} \frac{1}{N} \sum_{i=1}^N \nu_i^t \right\} ,
\]


(12)

where \( \{w_i^1\}_{i=1}^N \) and \( \{\nu_i^k\}_{i=1}^N \) are the (unnormalized) importance weights and the adjustment multipliers, respectively.

It may not obvious to see why (12) is a natural estimator of the normalizing constant \( \hat{Z}_k \). However, a by now well known result is that this SMC-based estimator is unbiased.
This result is due to Del Moral (2004, Proposition 7.4.1) and, for the special case of inference in state-space models, it has also been established by Pitt et al. (2012). To make this result more accessible, we provide a self-contained and direct proof of this result in the supplementary material, generalizing the result by Pitt et al. (2012).

**Proposition 1.** For any \( N \geq 1 \) and any \( k \in \{1, \ldots, K\} \)
\[
\mathbb{E}[\hat{Z}_k^N] = Z_k.
\]

**Proof.** See the supplementary material. \( \square \)

Since \( Z_K = Z \), we thus obtain an estimator of the partition function of the PGM at iteration \( K \) of the sampler. Besides from being unbiased, this estimator is of course also consistent and asymptotically normal; see Del Moral (2004).

### 4. Particle MCMC and partial blocking

Two shortcomings of SMC are: (i) it does not solve the parameter learning problem, and (ii) the quality of the estimates of marginal distributions
\[
p(X_{L_K}) = \int \gamma_K(X_{L_K})dX_{L_K|L_K}
\]
deteriorates for \( k \ll K \) due to the fact that the particle trajectories degenerate as the particle system evolves (see e.g., Doucet & Johansen (2011)). Many methods have been proposed in the literature to address these problems; see e.g. Lindsten & Schön (2013) and the references therein. Among these, the recently proposed particle MCMC (PMCMC) framework, introduced by Andrieu et al. (2010), plays a prominent role. In this section we will make use of the previously derived SMC algorithm within PMCMC to construct efficient, high-dimensional MCMC kernels for PGMs. This allows us to simulate, jointly, blocks of variables in the PGM using an MCMC scheme. This approach can then be coupled with standard MCMC procedures for learning unknown parameters of the model.

#### 4.1. Particle MCMC

PMCMC makes use of SMC to construct Markov kernels that are reversible w.r.t. to the target distribution of the sampler, in our case the full joint distribution \( p(X_Y) \). These methods can be divided into two classes, those based on particle independent Metropolis-Hastings and those based on particle Gibbs (PG). We shall focus on the latter class and, in particular, on the PG sampler with ancestor sampling (PGAS) by Lindsten et al. (2012), since this has been found to make efficient use of the simulated particles. It should be noted, however, that the proposed SMC sampler can be used together with other types of PMCMC samplers as well.

The SMC sampler used by PGAS differs from Algorithm 1 in that one particle, denoted as \( X_{L_K}^i \), is specified a priori. This particle, which serves as a type of reference for the PGAS sampler, is taken into account by simulating only \( N-1 \) particles in the usual way. The \( N \)th particle is then set deterministically according to the target. At the initialization, we thus simulate independently \( \{X_{L_1}^i\}_{i=1}^{N-1} \) with \( X_{L_1}^i \sim r_1(\cdot) \) and set \( X_{L_1}^N = X_{L_1}^i \). We then compute importance weights \( w_i \) for all particles, \( i = 1, \ldots, N \).

Analogously, at any consecutive iteration \( k \), we sample the first \( N-1 \) ancestor indices and particle increments \( \{a_k^i, s_k^i\}_{i=1}^{N-1} \) according to (6) and (7). The \( N \)th particle increment is set deterministically as \( s_k^N = X_{L_k}^r = X_{L_k}^i \). The ancestor index \( a_k^N \) corresponding to \( \xi_k^N \) is simulated in a so called ancestor sampling step, thereby assigning an artificial history for the reference particle. This step will be detailed in the subsequent section. Finally, importance weights are computed for all the particles according to (9).

After a complete pass of the above procedure, a particle \( X_{L_K}^i \) is sampled from among the weighted particles \( \{X_{L_K}^i, w_i^i\}_{i=1}^N \) according to
\[
\mathbb{P}(X_{L_K} = X_{L_K}^i) = \frac{w_i^i}{\sum_{i=1}^N w_i^i}, \quad i = 1, \ldots, N.
\]

This procedure thus associates each point \( X_{L_K}^i \subseteq X_Y \) to a probability distribution on \( X_Y \), thereby defining a Markov kernel—the PGAS kernel—on the space of latent variables of the PGM. Specifically, this Markov kernel is given by
\[
P_N(X_{L_K}^i, dX_{L_K}) := \mathbb{E}_{X_{L_K}} \left[ \frac{\sum_{i=1}^N w_i^i \delta_{X_{L_K}^i} (dX_{L_K})}{\sum_{i=1}^N w_i^i} \right].
\]

As an effect of the reference particle, the PGAS kernel is reversible w.r.t. the target distribution \( p(X_{L_K}) \). Additionally, the kernel is under mild assumptions uniformly ergodic. Quite remarkably, these properties hold for any number of particles \( N \geq 2 \) (Lindsten et al., 2012). Despite the fact that we cannot compute the expectation in (14) explicitly, we can simulate from the PGAS Markov kernel; we simply run the SMC-like procedure as outlined above. This implies that we can embed the PGAS kernel in an MCMC scheme to simulate from the latent variables, as well as from possible unknown parameters, of the model.

Simulating all the latent variables \( X_{L_K} \) jointly is useful since, in general, this will reduce the autocorrelation when compared to simulating the variables \( x_j \) one at a time (Robert & Casella, 2004). We can also use PGAS to construct an algorithm in between these two approaches, by simulating blocks of variables (but not necessarily all of them) jointly. We discuss this in Section 4.3 below.
4.2. Ancestor sampling

To implement the PGAS sampling procedure, it remains to detail the ancestor sampling step. At each iteration \( k \geq 2 \), this step amounts to generating a value for the ancestor index \( a_k^N \) corresponding to the reference particle. Implicitly, this assigns an artificial history for the “remaining” part of the reference particle \( X_{L_K \setminus L_{k-1}} \), by selecting one of the particles \( \{ X^i_{L_{k-1}} \}_{i=1}^N \) as its ancestor. This results in a complete assignment for the collection of latent variables of the PGM:

\[
\tilde{X}_{L_K} := \{ X^i_{L_{k-1}} \cup X^i_{L_K \setminus L_{k-1}} \} \in X_V. \tag{15}
\]

As shown by Lindsten et al. (2012), the probability distribution from which \( a_k^N \) should be sampled in order to ensure reversibility of the PGAS kernel w.r.t. \( \gamma_k \) is given by,

\[
P( a_k^N = i ) \propto w_{k-1}^i \frac{\gamma_k( \tilde{X}_{L_K}^i )}{\gamma_{k-1}(X_{L_{k-1}}^i)}. \tag{16}
\]

This expression can be understood as an application of Bayes’ theorem, where \( w_{k-1}^i \) is the prior probability of particle \( X^i_{L_{k-1}} \) and the ratio between the target densities is the unnormalized likelihood of \( X_{L_K \setminus L_{k-1}} \), conditionally on \( X_{L_{k-1}}^i \).

To derive an explicit expression for the ancestor sampling probabilities in our setting, note first that by (5),

\[
\frac{\gamma_K( X_{L_K} )}{\gamma_{k-1}(X_{L_{k-1}})} = \prod_{j=1}^K \psi_j( X_j^i ). \tag{17}
\]

Now, let \( A_k \) be the index set of factors \( \psi_j, j \geq k \) for which any of the variables \( X_{L_{k-1}} \) is in the domain of \( \psi_j \); formally \( A_k := \{ j : k \leq j \leq K, L_{k-1} \cap I_j \neq \emptyset \} \).

It follows that any factor \( \psi_j \) for which \( j \notin A_k \) is independent of \( X_{L_{k-1}} \). Consequently, we can write (16) as

\[
P( a_k^N = i ) \propto w_{k-1}^i \prod_{j \in A_k} \psi_j( \tilde{X}_{I_j}^i ), \tag{19}
\]

where \( \tilde{X}_{I_j}^i \) is a subset of the variables in (15). In fact, the index set \( A_k \) corresponds exactly to the factors \( \psi_j \) that depend, explicitly, both on the particle \( X^i_{L_{k-1}} \) and on the reference particle \( X^i_{L_K \setminus L_{k-1}} \) (though some of their respective components). Indeed, it is only these factors that hold any information about the likelihood of \( X^i_{L_K \setminus L_{k-1}} \) given \( X^i_{L_{k-1}} \).

The expression (19) is interesting, since it shows that the computational complexity of the ancestor sampling step will depend on the cardinality of the set \( A_k \). While this will depend both on the structure of the graph and on the ordering of the factors, it will for many models of interest be of a lower order than the cardinality of \( V \).

4.3. Partial blocking

The PGAS method corresponds to a blocking of the latent variables of the PGM in an MCMC scheme. However, an interesting property of the PGAS framework is that it can also be used for partial blocking of the latent variables. This implies that we simulate blocks of latent variables jointly, but the blocks can themselves be subsets of the collection of latent variables of the PGM.

Let \( \{ \gamma^m, m \in \{ 1, \ldots, M \} \} \) be a partition of \( V \). Ideally, a Gibbs sampler for the joint distribution \( p(X_V) \) could then be constructed by simulating, using a systematic or a random scan, from the conditional distributions

\[
p(X_{\gamma^m} | X_{\gamma \setminus \gamma^m}) \quad \text{for} \quad m = 1, \ldots, M. \tag{20}
\]

Note that, if we set \( M = |V| \) and \( \gamma^1 = \gamma \), this scheme reduces to a standard one-at-a-time Gibbs sampler. On the other extreme, if we set \( M = 1 \) and \( \gamma^1 = \emptyset \), we obtain a fully blocked sampler which aims to sample directly from the full joint distribution \( p(X_V) \).

From (1) it follows that the conditional distributions (20) can be expressed as

\[
p(X_{\gamma^m} | X_{\gamma \setminus \gamma^m}) \propto \prod_{C \in \gamma^m} \psi_C(X_C), \tag{21}
\]

where \( \gamma^m = \{ C : C \cap \gamma^m \neq \emptyset \} \). While it is in general not possible to sample exactly from these conditionals, we can make use of the PGAS procedure to facilitate a partially blocked Gibbs sampler for a PGM. Recall that the PGAS kernel (14) targets the full joint distribution \( p(X_V) \). However, by letting \( p(X_{\gamma^m} | X_{\gamma \setminus \gamma^m}) \) be the target distribution for the underlying SMC sampler, we obtain an MCMC kernel \( P_N^m \) that is reversible w.r.t. this conditional distribution. Consequently, we can construct an MCMC scheme as follows: with \( X_{\gamma^m} \) being the current state of the Markov chain, update block \( m \) by sampling

\[
X_{\gamma^m} \sim P_N^m(X_{\gamma \setminus \gamma^m} | X_{\gamma^m}, \cdot). \tag{22}
\]

Here we have indicated explicitly in the notation that the PGAS kernel for the conditional distribution \( p(X_{\gamma^m} | X_{\gamma \setminus \gamma^m}) \) depends on both \( X_{\gamma \setminus \gamma^m} \) (which is considered to be fixed throughout the sampling procedure) and on \( X_{\gamma^m} \) (which will define the reference particle).

Partial blocking is particularly useful when the subgraphs induced by the vertex sets \( \gamma^m \) for \( m = 1, \ldots, M \) are such that we can easily make a sequential decomposition of each of these graphs. This is the case, for instance, when the partition is chosen in such a way that all the induced subgraphs are chains.
5. Experiments

In this section we profile the proposed SMC sampler on two examples to illustrate the merits of our approach. The first example is a standard square lattice Gaussian Markov random field (MRF), which shows that our proposed method can significantly decrease correlations between samples in an MCMC scheme. In the second experiment we consider the two-dimensional classical XY model with non-Gaussian interactions and use the proposed SMC algorithm to estimate the partition function.

5.1. Gaussian MRF

Consider a square lattice Gaussian MRF of size $10 \times 10$, given by the relation

$$
p(X_V, Y_V) \propto \prod_{i \in V} e^{\frac{1}{2\sigma_i^2} (x_i - y_i)^2} \prod_{(i,j) \in E} e^{\frac{1}{2\sigma_{ij}^2} (x_i - x_j)^2},
$$

with latent variables $X_V = \{x_1, \ldots, x_{100}\}$ and measurements $Y_V = \{y_1, \ldots, y_{100}\}$. The graphical representation of the latent variables in this model is shown in Figure 3.

The measurements $Y_V$ where simulated from the model with $\sigma_i = 1$ and $\sigma_{ij} = 0.1$. Given these measurement, we seek the posterior distribution $p(X_V \mid Y_V)$. We run four different MCMC samplers to simulate from this distribution; the proposed (fully blocked) PGAS, the proposed PGAS with partial blocking, a standard one-at-a-time Gibbs sampler, and the tree-sampler proposed by Hamze & de Freitas (2004). For the PGAS algorithms we use $N = 50$ particles. The tree-sampler exploits the fact that the model is Gaussian and it can thus not be used for arbitrary (non-Gaussian or non-discrete) graphs. By partitioning the graph into disjoint trees (in our case, chains) for which exact inference is possible, the tree-sampler implements an “ideal” partially blocked Gibbs sampler. PGAS with partial blocking can thus be seen as an SMC-based version of the tree-sampler. See Figure 3 for the ordering in the PGAS algorithm and the blocking used for treesampling and PGAS with partial blocking, corresponding to a partition of the graph into two chains. The variables are numbered $1, \ldots, 100$ from top to bottom, left to right and $L_k$ is taken as the $k$ first indices of $L_k = \{1, \ldots, 10, 20, 19, \ldots, 11, 21, 22, \ldots, 100, 99, \ldots, 91\}.$

In Figure 4 we can see the empirical autocorrelation functions (ACF) centered around the true posterior mean for variable $x_{e2}$ (selected randomly from among $X_V$). Similar results hold for all the variables of the model. Due to the strong interaction between the latent variables; the samples generated by the standard Gibbs sampler are strongly correlated. Tree-sampling and PGAS with partial blocking show nearly identical gains compared to Gibbs. This is interesting, since it suggest that simulating from the SMC-based PGAS kernel can be almost as efficient as exact simulation, even using a moderate number of particles. We emphasize that the PGAS kernels leave their respective target distributions invariant, i.e. the limiting distributions is the same for all MCMC schemes. The fully blocked PGAS algorithm achieves the best ACF, dropping off to zero considerably faster than for the other methods. This is not surprising since this sampler simulates all the latent variables jointly which reduces the autocorrelation, in particular when the latent variables are strongly dependent.

However, this improvement in autocorrelation comes at a cost. For the fully blocked PGAS kernel, the maximal cardinality of the set $A_k$ (see (18)) is 10 (one full row of variables). For the partially blocked PGAS kernel, on the other hand, $|A_k| \equiv 1$ since the variables in each block form a chain. This implies that the fully blocked PGAS sampler is an order of magnitude more computationally involved than the partially blocked PGAS sampler. This trade-off between autocorrelation and computational efficiency has to be taken into account when deciding which algorithm that is most suitable for any given problem.
5.2. Classical XY model

The classical XY model, see e.g. (Kosterlitz & Thouless, 1973) and references therein, is a member in the family of \textit{n-vector} models used in statistical mechanics. It can be seen as a generalization of the well known Ising model with a two-dimensional electromagnetic spin. The spin vector is described by its angle \( x \in (-\pi, \pi] \). We will consider a square lattice of sites described by their spin angle.

The full joint PDF of the classical XY model is given by

\[
 p(X_Y) \propto e^{-\beta H(X_Y)},
\]

where \( \beta \) is the inverse temperature and \( H(X_Y) \)—the Hamiltonian—is a sum of pair-wise interaction described by

\[
 H(X_Y) = - \sum_{(i,j) \in E} J_{ij} \cos (x_i - x_j),
\]

where the \( J_{ij} \)'s are parameters describing interactions between the different sites.

By adding factors in the same fashion as described in the Gaussian MRF example, see Figure 3, we can choose our sequence of intermediate target distributions as

\[
 \gamma_k(X_{\mathcal{L}_k}) \propto \gamma_{k-1}(X_{\mathcal{L}_{k-1}}) \prod_{i \in \mathcal{N}_k} e^{\beta J_{k,i} \cos (x_k - x_i)}
\]

\[
 \propto \gamma_{k-1}(X_{\mathcal{L}_{k-1}}) e^{\kappa(X_{\mathcal{L}_{k-1}}) \cos (x_k - \mu(X_{\mathcal{L}_{k-1}}))}, \]

where \( \mathcal{N}_k = \{i : (k, i) \in \mathcal{E}\} \cap \mathcal{L}_{k-1} \) denotes the set of neighbours to variable \( k \) in \( \mathcal{L}_{k-1} \). The quantities \( \mu(X_{\mathcal{L}_{k-1}}) \) and \( \kappa(X_{\mathcal{L}_{k-1}}) \) follow from elementary trigonometric operations. From the above expression we note that, conditionally on \( X_{\mathcal{L}_{k-1}} \), the variable \( x_k \) is von Mises distributed under \( \gamma_k \), with \( X_{\mathcal{L}_{k-1}} \)-dependent mean \( \mu \) and dispersion \( \kappa \). This implies that we can employ full adaption of the proposed SMC sampler. This is accomplished by choosing as proposal distribution \( r_k(x_k | X_{\mathcal{L}_{k-1}}) \) the aforementioned von Mises distribution and by choosing as adjustment weights the corresponding normalizing constants:

\[
 \nu(X_{\mathcal{L}_{k-1}}) = 2\pi I_0(\kappa(X_{\mathcal{L}_{k-1}}))
\]

where \( I_0 \) is the modified Bessel function of order 0.

We use the fully adapted SMC sampler to estimate the partition function of the classical XY model according to (12). Note that this is done by a single run of the SMC algorithm, i.e., we do not couple the sampler with MCMC as in the previous example. In Figure 5 and 6 we present the results in the form of box-plots over 10 independent runs of the algorithm for the 10 \( \times \) 10 and 50 \( \times \) 50 square lattice structures, respectively. Interaction parameters \( J_{ij} \) were independently drawn from a standard uniform distribution, i.e. \( J_{ij} \sim \mathcal{U}(0, 1) \), and \( \beta \) was set to 2 corresponding to a low temperature.

The method performs well on the 10 \( \times \) 10 graph already with \( N = 1000 \). The negative bias that we experience for small \( N \) is due to the fact that we are plotting \( \log \hat{Z}^N \) and not \( \hat{Z}^N \) (which is unbiased, see Proposition 1). This bias diminishes at a rate 1/\( N \) and for large \( N \) it is dominated by the variance. In the second example, with 25 times as many variables, we need more particles to obtain an accurate estimate of the partition function. This is in agreement with the rule-of-thumb for SMC samplers, to scale \( N \) at least linearly with the number of latent variables. As we have previously mentioned, there are not many methods available for estimating the partition function in a loopy, non-Gaussian, and non-discrete graphical model. With this in mind, we believe that the proposed method provides promising results at a reasonably low computational cost.

6. Discussion

We have proposed several SMC-based inference methods for graphical models of arbitrary topologies and with discrete or continuous random variables. We showed the efficacy of our methods on two types of problems, inferring the latent variables and estimating the partition function of a PGM. It is also possible to combine our method with
a tempering effect as used for example in hot coupling by Hamze & de Freitas (2005).

We show in our work that for a rectangular lattice structure it is straightforward to choose a sequential decomposition of the PGM that appears to work well in practice. However, these decompositions are neither unique nor is the performance of the sampler invariant to this choice for a finite amount of computational effort. To choose a sequential decomposition, aspects such as accuracy, computational efficiency, and simplicity of implementation have to be taken into account. For the general case this choice is a difficult problem that demands further study. When we use partial blocking and construct blocks of chains, however, the problem essentially collapses to a smoothing problem in a nonlinear/non-Gaussian state-space model which is a well studied problem, see e.g. Lindsten & Schööhn (2013).

References


Sequential Monte Carlo methods for graphical models – Supplementary material

Anonymous Author(s)
Affiliation
Address
e-mail

Abstract

This supplementary material contains a simple and direct proof of the unbiasedness of the partition function estimator \( \hat{Z}_k^N \), stated as Proposition 1 in the main manuscript. It should be noted, however, that this result is not new. It has previously been established in a general setting by Del Moral (2004, Proposition 7.4.1) and, additionally, by Pitt et al. (2012) who provide a more accessible proof for the special case of state-space models. Our proof is similar to that of Pitt et al. (2012), but generalized to the PGM setting that we consider.

1 Proof of Proposition 1

Recall that we use the convention \( \xi_k = X_{k-1} \). Define recursively the functions \( f_k(X_{L_k}) \equiv 1 \) and,

\[
f_{\ell}(X_{L_{\ell}}) = \frac{\int f_{\ell+1}(X_{L_{\ell+1}}) \gamma_{\ell}(X_{L_{\ell+1}}) d\xi_{\ell+1}}{\gamma_{\ell}(X_{L_{\ell}})} \quad (1)
\]

for \( \ell = k-1, k-2, \ldots, 1 \). Let

\[
Q_\ell = \left( \frac{1}{N} \sum_{i=1}^{N} w_{\ell} \xi_{\ell}(X_{L_{\ell}}) \right) \left\{ \prod_{m=1}^{\ell-1} \frac{1}{N} \sum_{i=1}^{N} \nu_{i}^{m}w_{m}^{i} \right\},
\]

for \( \ell \in \{1, \ldots, k\} \). Note that, by construction, \( Q_k = \hat{Z}_k^N \). Let \( \mathcal{F}_\ell \) be the filtration generated by the particles simulated up to iteration \( \ell \):

\[
\mathcal{F}_\ell := \sigma(\{X_{L_{m_i}}, w_{m_i}^{i}\}_{i=1}^{N}, m = 1, \ldots, \ell).
\]

Lemma 1. The sequence \( \{Q_\ell, \ell = 1, \ldots, k\} \) is an \( \mathcal{F}_\ell \)-martingale.

Proof. Consider,

\[
\mathbb{E}[Q_\ell | \mathcal{F}_{\ell-1}] = \mathbb{E}\left[ w_{\ell} f_{\ell}(X_{L_{\ell}}) | \mathcal{F}_{\ell-1} \right] \left\{ \prod_{m=1}^{\ell-1} \frac{1}{N} \sum_{i} \nu_{i}^{m}w_{m}^{i} \right\}.
\]
Using the definition of the weight function (Equation (9) in the main document) we have,

\[ E \left[ W_\ell(X^1_{L_\ell}) f_\ell(X^1_{L_\ell}) \mid F_{\ell-1} \right] = \sum_{i=1}^{N} \int W_\ell(\{X^i_{L_{\ell-1}} \cup \xi_\ell\}) f_\ell(\{X^i_{L_{\ell-1}} \cup \xi_\ell\}) \frac{\nu_{\ell-1}^i w_{\ell-1}^i}{\nu_{\ell-1}^i w_{\ell-1}^i} r_\ell(\xi_\ell|X^i_{L_{\ell-1}}) d\xi_\ell \]

\[ = \frac{1}{\sum l \nu_{\ell-1}^l w_{\ell-1}^l} \sum_{i=1}^{N} \left( \nu_{\ell-1}^i w_{\ell-1}^i \int \frac{\gamma_\ell(\{X^i_{L_{\ell-1}} \cup \xi_\ell\}) f_\ell(\{X^i_{L_{\ell-1}} \cup \xi_\ell\}) d\xi_\ell}{\gamma_{\ell-1}(X^i_{L_{\ell-1}}) \nu_{\ell-1}^i (X^i_{L_{\ell-1}})} \right) \]

\[ = \frac{1}{\sum l \nu_{\ell-1}^l w_{\ell-1}^l} \sum_{i=1}^{N} \left( w_{\ell-1}^i f_\ell-1(X^i_{L_{\ell-1}}) \right) . \]

Hence, we get

\[ E[Q_\ell \mid F_{\ell-1}] = \sum_i \left( w_{\ell-1}^i f_\ell-1(X^i_{L_{\ell-1}}) \right) \frac{1}{N} \left\{ \prod_{m=1}^{\ell-2} \frac{1}{N} \sum_i \nu_m^i w_m^i \right\} = Q_{\ell-1}. \]

\[ \square \]

It follows that

\[ \mathbb{E}[\hat{Z}_k^N] = \mathbb{E}[Q_k] = \mathbb{E}[Q_1] = \int W_1(X_{L_1}) f_1(X_{L_1}) r_1(X_{L_1}) dX_{L_1} = \int \gamma_1(X_{L_1}) f_1(X_{L_1}) dX_{L_1}. \]

However, from the definition in \([1]\) we have that

\[ \int \gamma_3(X_{L_1}) f_1(X_{L_1}) dX_{L_1} = \int \int \gamma_2(X_{L_2}) f_2(X_{L_2}) dX_{L_2} \]

\[ = \cdots = \int \cdots \int \gamma_k(X_{L_k}) f_k(X_{L_k}) dX_{L_k} = Z_k. \]

\[ \square \]

References
