Journal of Mathematical Psychology 73 (2016) 140–152

Contents lists available at ScienceDirect

Journal of Mathematical Psychology

journal homepage: www.elsevier.com/locate/jmp

A tutorial on particle filters

Maarten Speekenbrink

Department of Experimental Psychology, University College London, Gower Street, London WC1E 6BT, England, United Kingdom

HIGHLIGHTS

- Introduces the reader to particle filters and sequential Monte Carlo (SMC).
- SMC allows Bayesian inference in complex dynamic models common in psychology.
- Provides an in depth discussion of (sequential) importance sampling.
- Highlight advantages and issues with SMC.
- All examples can be replicated with provided R code.

ARTICLE INFO

Article history:
Received 21 September 2015
Received in revised form
9 May 2016

Keywords:
Particle filter
Sequential Monte Carlo
State-space model
Sequential Bayesian inference

ABSTRACT

This tutorial aims to provide an accessible introduction to particle filters, and sequential Monte Carlo (SMC) more generally. These techniques allow for Bayesian inference in complex dynamic state-space models and have become increasingly popular over the last decades. The basic building blocks of SMC – sequential importance sampling and resampling – are discussed in detail with illustrative examples. A final example presents a particle filter for estimating time-varying learning rates in a probabilistic category learning task.

Particle filters, and sequential Monte Carlo (SMC) techniques more generally, are a class of simulation-based techniques which have become increasingly popular over the last decades to perform Bayesian inference in complex dynamic statistical models (e.g., Doucet, de Freitas, & Gordon, 2001b; Doucet & Johansen, 2011). Particle filters are generally applied to so-called filtering problems, where the objective is to estimate the latent states of a stochastic process on-line, such that, after each sequential observation, the state giving rise to that observation is estimated. For instance, in a category learning task, we might want to infer how people use the features of objects to categorize them. Due to learning, we would expect their categorization strategy to change over time. Traditionally, a formal learning model such as ALCOVE (Kruschke, 1992) would be used for this purpose, which describes how feedback on their categorization decisions affects people's momentary strategy. However, these models usually assume a deterministic updating process, which may be too restrictive. Ideally, we would like to estimate someone's strategy – which we can view as the latent state of their decision process – from trial to trial whilst allowing for stochastic transitions between states. Estimating the current categorization strategy is a difficult task, however, as a single categorization decision at each point in time provides relatively little information about people's complete categorization strategy, i.e. their potential categorizations of all possible stimuli. Assuming trial-to-trial changes to a state (strategy) are noisy but relatively small, we may however be able to gain some insight into the current state from all previous categorization decisions someone made. This filtering problem is generally not analytically tractable; analytical results are only available for the restricted class of linear Gaussian state-space models. As particle filters are applicable to the much broader class of non-linear non-Gaussian state-space models, they open up interesting possibilities to study a broad range of dynamic processes in psychology.

A graphical representation of a generic particle filter (see Section 4.3) is given in Fig. 1. Particle filters operate on a set of randomly sampled values of a latent state or unknown parameter. The sampled values, generally referred to as "particles", are propagated over time to track the posterior distribution of the state or parameter at each point in time. Each particle is assigned a weight in relation to its posterior probability. To increase their accuracy, SMC techniques resample useful particles from the set according to these weights. This resampling introduces interaction

E-mail address: m.speekenbrink@ucl.ac.uk.
URL: http://www.ucl.ac.uk/speekenbrink-lab/.

http://dx.doi.org/10.1016/j.jmp.2016.05.006
0022-2496/© 2016 Elsevier Inc. All rights reserved.

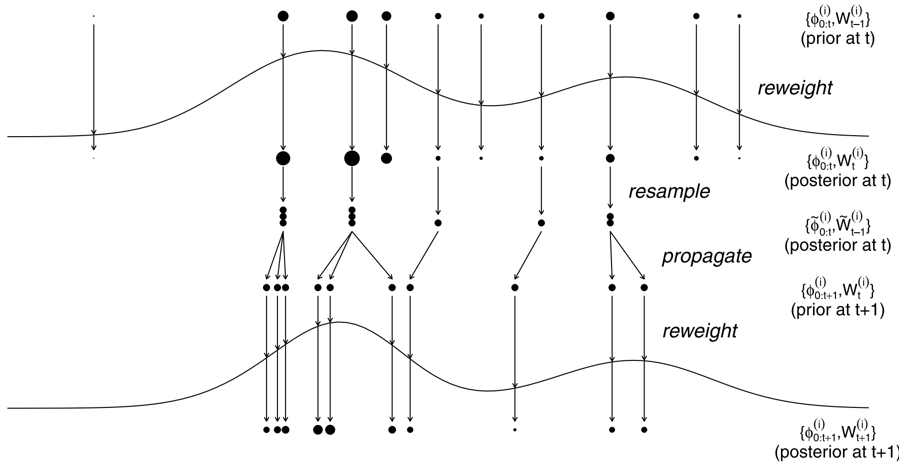
$\{\phi_{0:t}^{(i)}, W_{t-1}^{(i)}\}$
(prior at t)
reweight
$\{\phi_{0:t}^{(i)}, W_{t}^{(i)}\}$
(posterior at t)
resample
$\{\tilde{\phi}_{0:t}^{(i)}, \tilde{W}_{t-1}^{(i)}\}$
(posterior at t)
propagate
$\{\phi_{0:t+1}^{(i)}, W_{t}^{(i)}\}$
(prior at t+1)
reweight
$\{\phi_{0:t+1}^{(i)}, W_{t+1}^{(i)}\}$
(posterior at t+1)

Fig. 1. Schematic representation of a generic particle filter (after Doucet et al., 2001a). Standing at time t, we have a set of weighted particles $\{\phi_{0:t}^{(i)}, W_{t-1}^{(i)}\}$ representing the prior distribution at t. Each particle $\phi_{0:t}^{(i)}$ is a multidimensional variable which represents the whole path of the latent state from time 0 up to the current time point t, such that each dimension represents the value of the state at a particular time point. The location of the dots in the graph reflect $\phi_t^{(i)}$, the value of the state at the current time point, i.e. the dimension of each particle reflecting the current state. The size of each dot reflects the weight $W_{t-1}^{(i)}$ ("prior at t"). In the *reweight* step, the weights are updated to $W_t^{(i)}$ partly as a function of $p(y_t|\phi_t^{(i)})$, the likelihood of observation y_t according to each sampled state value $\phi_t^{(i)}$ (solid line). The resulting set $\{\phi_{0:t}^{(i)}, W_t^{(i)}\}$ of weighted particles approximates the posterior distribution ("posterior at t") of the latent state paths. The *resampling* step duplicates values $\phi_{0:t}^{(i)}$ with high weights $W_t^{(i)}$, and eliminates those with low weights, resulting in the set of uniformly weighted particles $\{\tilde{\phi}_{0:t}^{(i)}, \tilde{W}_t^{(i)} = 1/N\}$ which is approximately distributed according to the posterior (second "posterior at t"). In the *propagate* step, values of states $\phi_{t+1}^{(i)}$ at the next time point are sampled and added to each particle to account for state transitions, forming a prior distribution for time $t+1$ ("prior at $t+1$"). Thus, at each new time point, the particles grow in dimension because the whole path of the latent state now incorporates the new time point as well. The particles are then reweighted in response to the likelihood of the new observation y_{t+1} to approximate the posterior distribution at $t+1$ ("posterior at $t+1$"), etc.

between the particles, and the term "interacting particle filters" was coined by Del Moral (1996), who showed how the method relates to techniques used in physics to analyze the movement of particles.

Particle filters have successfully solved difficult problems in machine learning, such as allowing robots to simultaneously map their environment and localize their position within it (Montemerlo, Thrun, Koller, & Wegbreit, 2002), and the automated tracking of multiple objects in naturalistic videos (Isard & Blake, 1998; Nummiaro, Koller-Meier, & Gool, 2003). More recently, particle filters have also been proposed as models of human cognition, for instance how people learn to categorize objects (Sanborn, Griffiths, & Navarro, 2010), how they detect and predict changes (Brown & Steyvers, 2009) as well as make decisions (Yi, Steyvers, & Lee, 2009) in changing environments.

The aim of this tutorial is to provide readers with an accessible introduction to particle filters and SMC. We will discuss the foundations of SMC, sequential importance sampling and resampling, in detail, using simple examples to highlight important aspects of these techniques. We start with a discussion of importance sampling, which is a Monte Carlo integration technique which can be used to efficiently compute expected values of random variables, including expectations regarding the posterior probabilities of latent states or parameters. We will then move on to sequential importance sampling, an extension of importance sampling which allows for efficient computation in sequential inference problems. After introducing resampling as a means to overcome some problems in sequential importance sampling, we have all the ingredients to introduce a generic particle filter. After discussing limitations and extensions of SMC, we will conclude with a more complex example involving the estimation of time-varying learning rates in a probabilistic category learning task.

1. Importance sampling

Importance Sampling (IS) is a Monte Carlo integration technique. It can be used to efficiently solve high-dimensional integration problems when analytical solutions are difficult or unobtainable. In statistics, it is often used to approximate expected values of random variables, which is what we will focus on here. If we have a sample of realizations of a random variable Y, we can estimate the expected value by computing a sample average. We do this when we have data from experiments, and it is also the idea behind basic Monte Carlo integration. Importance sampling is based on the same idea, but rather than sampling values from the true distribution of Y, values are sampled from a different distribution, called the importance distribution. Sampling from a different distribution can be useful to focus more directly on the estimation problem at hand, or if it is problematic to sample from the target distribution. To correct for the fact that the samples were drawn from the importance distribution and not the target distribution, weights are assigned to the sampled values which reflect the difference between the importance and target distribution. The final estimate is then a weighted average of the randomly sampled values.

Suppose we wish to compute the expected value of an arbitrary function f of a random variable Y which is distributed according to a probability distribution p:

$$\mathbb{E}_p[f(Y)] \triangleq \int f(y)p(y)\, dy.$$

This is just the usual definition of an expected value (we use $\mathbb{E}_p$ to denote an expectation of a random variable with distribution p, and the symbol $\triangleq$ to denote 'is defined as'). The function f depends on what we want to compute. For instance, choosing $f(y) = y$ would result in computing the mean of Y, while choosing $f(y) = (y - \mathbb{E}_p[f(Y)])^2$ would result in computing the variance of Y. It is often not possible to find an analytical solution to the integral above, in which case we have to turn to some form of numerical approximation. A basic Monte Carlo approximation is to draw a number of independent samples from p and then compute a sample average from these random draws:

Algorithm 1. Basic Monte Carlo integration for an expected value $\mathbb{E}_p[f(Y)]$