Sequential Monte Carlo Methods

Andrea Becsek, Mauro Camara, Doug Corbin

May 17, 2020

Contents

1 Introduction

To be completed once we have a better understanding of SMC methods.

2 Sampling Methods

Suppose that, for some random variable $\boldsymbol{X}$ with probability density function p, we wish to compute the expectation

$$\mathbb{E}[f(\boldsymbol{X})] = \int f(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{1}$$

for some function f. In many cases, f and p are such that (1) cannot be computed analytically, therefore we must look for suitable approximations. Consider a random sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ drawn from p, and compute the sample mean

$$\bar{f}_m^* := \frac{1}{m}\sum_{i=1}^{m} f(\boldsymbol{x}_i). \tag{2}$$

From the weak law of large numbers, it follows that

$$\lim_{m\to\infty} \mathbb{P}(|\bar{f}_m^* - \mathbb{E}[f]| > \epsilon) = 0 \tag{3}$$

for any $\epsilon > 0$. This result forms the basis of sampling/monte carlo methods, which approximate integrals of the form (1) by a discrete sum of samples drawn from p. However, in practice we are unable to sample directly from p, thus we require sampling methods. In this section we cover sampling methods which are most relevant to Sequential Monte Carlo (SMC) methods, however a general overview of sampling methods can be found in [1, Chapter 11].

2.1 Importance Sampling

The key idea of importance sampling is to approximately sample from p by first drawing samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$ from some *trial distribution* $q()$, then computing

$$\bar{f}_m := \sum_{i=1}^{m} w_i f(\boldsymbol{x}_i) \tag{4}$$

where $\boldsymbol{w} = (w_1, \dots, w_m)$ are weights which correct for the bias introduced by sampling from q instead of p. We start by deriving how we construct $\boldsymbol{w}$, then we proceed with an explanation of the trial distirbution $q()$ and how it is chosen.

Assuming $p(\boldsymbol{x})$, $q(\boldsymbol{x})$ are known up to a constant (i.e. $p(\boldsymbol{x}) = \tilde{p}(\boldsymbol{x})/C_p$ and $q(\boldsymbol{x}) = \tilde{q}(\boldsymbol{x})/C_q$), we can construct the following weights:

$$\begin{aligned} \mathbb{E}[f] &= \int f(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}x \\ &= \frac{C_q}{C_p} \int f(\boldsymbol{x}) \frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})} q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &\approx \frac{C_q}{C_p} \sum_{i=1}^{m} \frac{\tilde{p}(\boldsymbol{x}_i)}{\tilde{q}(\boldsymbol{x}_i)} f(\boldsymbol{x}_i) \\ &= \frac{C_q}{C_p} \sum_{i=1}^{m} w_i^* f(\boldsymbol{x}_i) \end{aligned} \tag{5}$$

Where we have defined $w_i^* = \tilde{p}(\boldsymbol{x}_i)/\tilde{q}(\boldsymbol{x}_i)$. The ratio of normalization constants can be approximated as

$$\begin{aligned} \frac{C_p}{C_q} &= \frac{1}{C_q} \int \tilde{p}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int \frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})} q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &\approx \sum_{i=1}^{m} w_i^*. \end{aligned} \tag{6}$$

Hence the final expression for w_i is written as

$$w_i = \frac{w_i^*}{\sum_{i=1}^{m} w_i^*} = \frac{\tilde{p}(\boldsymbol{x}_i)/\tilde{q}(\boldsymbol{x}_i)}{\sum_{i=1}^{m} \tilde{p}(\boldsymbol{x}_i)/\tilde{q}(\boldsymbol{x}_i)}. \tag{7}$$

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.