Practice 4

MLCG 2021-2022

Ricardo Marques

ricardo.marques@ub.edu

Universitat de Barcelona – Master in Artificial Intelligence (MAI)

Introduction

In this practice you will deal with importance sampling, a variance reduction typically used to accelerate convergence of Monte Carlo methods. In part I, you will use the appWorkbench to develop and evaluate a MCIS (Monte Carlo Importance Sampling) solution. Then, in part II, you will implement and evaluate the Bayesian Monte Carlo (BMC) equivalent of importance sampling (BMC IS) using the appWorkbench.

As an optional task, for each of the two parts, you might also provide an application to rendering.

Part I – MC Importance Sampling

As you saw in the theory class, adding importance sampling to MC integration is relatively straightforward. It basically amounts to replacing the uniform probability density used to sample the integrand by a non-uniform probability density that, we believe, mimics the shape of the integrand. As a result, the sampling effort should be concentrated in the zones of the integrand domain which, according to our beliefs, should have a larger value.

In this practice, when using the appWorkbench, we will consider the integral:

$$I = \int_{\Omega} f(\omega_i)\, p(\omega_i)\, \mathrm{d}\omega_i \,,$$

where $f(\omega_i)$ is unknown and $p(\omega_i)$ known (i.e., we know the value of $p(\omega_i)$ beforehand). For the sake of simplicity, let us assume that:

$$p(\omega_i) = \cos\theta_i \,,$$

with $\omega_i = (\theta_i, \phi_i)$ being a direction on the sphere. Since all the information we have about the shape of the integrand $f(\omega_i) \times p(\omega_i)$ is the fact that $p(\omega_i) = \cos\theta_i$, a sensible choice is to distribute our samples proportionally to $\cos\theta_i$, such that:

$$pdf(\omega_i) \propto \cos\theta_i$$

Note that, the mathematical expression of the CMC estimator that you used in Practice 2 does not change when using importance sampling (only the sample positions and the sample weights do).

Finally, to be able to collect samples of your integrand, you will have to specify a way to collect samples of the unknown function $f(\omega_i)$. This can be done by using an environment map, or any other function that you can sample. For the sake of simplicity, in the following of this practice, we will consider that the "unknown" function $f(\omega_i)$ is given by:

$$f(\omega_i) = \cos^3(\omega_i)$$

Once you have implemented the CMC IS estimator, you will be able to add it to your average error plot, yielding a plot roughly similar to the one shown in Figure 1:

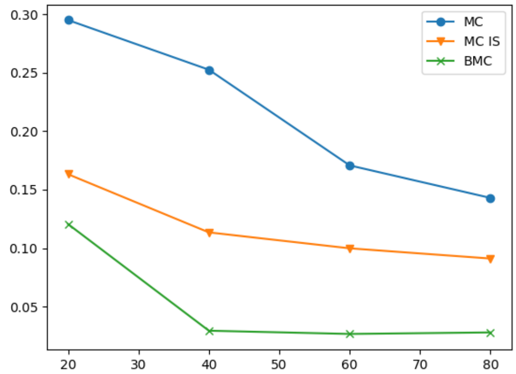

Figure 1. Average error as a function of the number of samples for Monte Carlo (MC), Monte Carlo Importance Sampling (MC IS) and Bayesian Monte Carlo. Note that, even using importance sampling, the MC IS method does not outperform the basic BMC method.

Optional step: Apply the CMC IS estimator to rendering using the appRenderer that you have developed in the previous practices. Note that, even if the step is not mandatory, to opt for the full Practice grade you must have it implemented.

Part II – BMC Importance Sampling

The goal of this second part is to implement the BMC equivalent to MC IS. In part I, we have considered that we know the mathematical expression of one of the terms of our integrand: $p(\omega_i) = \cos\theta_i$. We have used this prior information to distribute our samples such that they follow the shape of $p(\omega_i)$ by setting $pdf(\omega_i) \propto \cos\theta_i$. This is the MC way of (implicitly) introducing prior information in the system: we know one term ($p(\omega_i)$) of our integrand, so we try to concentrate the samples where this term takes a larger value.

In a Bayesian setting, as we have seen in the previous classes, this separation between what is known (and thus certain) from what is not (or only partially) known (and thus uncertain) is made in an explicit manner. Therefore, besides distributing the samples proportionally to $\cos\theta_i$, we will also explicitly set $p(\omega_i) = \cos(\theta_i)$ in the vector z of the Bayesian Monte Carlo equations, yielding:

$$z = \left(\int k(\omega_1, \omega)\ \cos(\theta_1)\ d\omega,\ \ \dots, \int k(\omega_N, \omega)\, p(\theta_N)\, d\omega \right)$$

Note that, with this approach only the term $f(\omega)$ is modeled by the Gaussian Process (and not the product $f(\omega)\,p(\omega)$, as you did in the previous lab). Therefore, the sample values must be collected from $f(\omega)$ only, instead of the whole integrand. Once you have the BMC IS implemented, you should have a result where the BMC IS approach outperforms consistently and for all number of samples the concurrent methods.

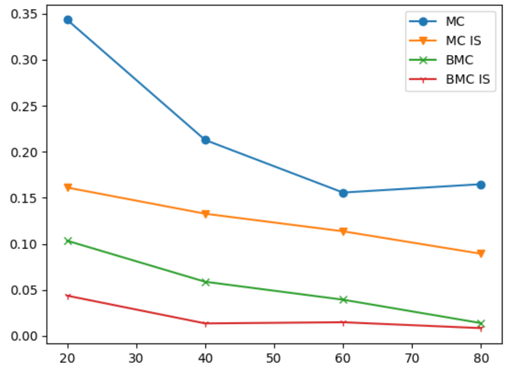

Figure 2. Final result of the average of the estimate error for all the 4 implemented methods.

Optional step: Apply the BMC IS estimator to rendering using the appRenderer that you have developed in the previous practices. Note that, even if the step is not mandatory, to opt for the full Practice grade you must have it implemented.