1 Theory

1.1 Two-Level Monte Carlo

1.1.1 Mean Estimator

Let $f(\boldsymbol{x}) \in L^2\left(\mathbb{R}^d, \phi(\boldsymbol{x}) d\boldsymbol{x}\right)$ be a function, whose input is distributed according to the multivariate d-dimensional random variable X, with probability density function (PDF) $\phi(\boldsymbol{x})$.

The aim is to estimate the mean $\mathbb{E}[f(X)]$, with the minimum number of evaluations of the function f. The simple MC estimator is

$$\mu_N = \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}_i), \tag{1}$$

where $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ is a set of samples drawn from $X_1, X_2, ..., X_N$ i.i.d. random variables. This estimator converges to the true mean

$$\lim_{N\to\infty} \mu_N = \mathbb{E}[f(X)], \quad \text{since } \operatorname{MSE}\left(\mu_N, \mathbb{E}[f(X)]\right) = \frac{\operatorname{Var}[f(X)]}{N}. \tag{2}$$

Therefore it is an unbiased estimator.

Now let $\widetilde{f}$ be a surrogate model that approximates f, but that is much cheaper to evaluate. We also assume that $\widetilde{f} \in L^2\left(\mathbb{R}^d, \phi(\boldsymbol{x}) d\boldsymbol{x}\right)$. Using this surrogate model to compute the sample mean, the estimator is

$$\widetilde{\mu}_N = \frac{1}{N} \sum_{i=1}^{N} \widetilde{f}(\boldsymbol{x}_i), \tag{3}$$

and the error is

$$\operatorname{MSE}\left(\widetilde{\mu}_N, \mathbb{E}[f(X)]\right) = \mathbb{E}^2\left[\widetilde{f}(X) - f(X)\right] + \frac{\operatorname{Var}\left[\widetilde{f}(X)\right]}{N}. \tag{4}$$

The first term is the bias, and the second term is the variance. The variance term quickly vanishes if we assume that $\widetilde{f}$ has a negligible runtime, and that thus $N \to \infty$. This method is biased.

Now we combine the surrogate model with f into a two-level estimator

$$\mu_{N,M} = \frac{1}{M} \sum_{i=1}^{M} \widetilde{f}(\boldsymbol{z}_i) + \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}_i) - \widetilde{f}(\boldsymbol{x}_i) = \widetilde{\mu}_M + \mu_N - \widetilde{\mu}_N, \tag{5}$$

where the two sets of samples $\boldsymbol{x}_1, \boldsymbol{x}_2, , ..., \boldsymbol{x}_N$ and $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_M$ are generated from i.i.d. random variables. Note that for this estimator we evaluate the surrogate model $N + M$ times, and the expensive model N times. This method is unbiased since

$$\operatorname{MSE}\left(\mu_{N,M}, \mathbb{E}[f(X)]\right) = \frac{\operatorname{Var}\left[\widetilde{f}(X)\right]}{M} + \frac{\operatorname{Var}\left[f(X) - \widetilde{f}(X)\right]}{N} \tag{6}$$

has variance terms, but no bias term.

1.2 Adaptive MFMC

Consider that we have N input-output samples $\boldsymbol{x}_1, f(\boldsymbol{x}_1), \boldsymbol{x}_2, f(\boldsymbol{x}_2) ..., \boldsymbol{x}_N, f(\boldsymbol{x}_N)$. From these, n will be used to train a surrogate model $f^{(n)}$, and $N - n$ will be used in the evaluation of the two-level estimator (5). The estimator becomes

$$\mu_{N,M}^{(n)} = \frac{1}{M} \sum_{i=1}^{M} f^{(n)}(\boldsymbol{z}_i) + \frac{1}{N-n} \sum_{i=n+1}^{N} f(\boldsymbol{x}_i) - f^{(n)}(\boldsymbol{x}_i) .$$

Then the MSE is

$$\text{MSE}\left(\mu_{N,M}^{(n)}, \mathbb{E}[f(X)]\right) = \frac{\text{Var}\left[f^{(n)}\right]}{M} + \frac{\text{Var}\left[f - f^{(n)}\right]}{N-n} \quad \text{with} \quad 0 \le n \le N .$$

Assume that $f^{(n)}$ has negligible runtime, and thus $M \to \infty$, then

$$\text{MSE} = \frac{V^{(n)}}{N-n} ,$$

where $V^{(n)} \stackrel{\text{def}}{=} \text{Var}[f - f^{(n)}]$. Then one needs to find the optimal n that minimises the MSE, which can be achieved by satisfying the following

$$\frac{d}{dn} \text{MSE} = \frac{\frac{dV^{(n)}}{dn}(N-n) + V^{(n)}}{(N-n)^2} \stackrel{!}{=} 0 \iff -\frac{dV^{(n)}}{dn} \stackrel{!}{=} \frac{V^{(n)}}{N-n} .$$

1.2.1 Algorithm for Adaptive MFMC

In the paper *Multifidelity Monte Carlo Estimation with Adaptive Low-Fidelity Models*, they estimate the upper bounds on $V^{(n)}$ (constants c_1 and α in the paper), and from those they use an analytical formula to minimise the MSE. This requires a numerical estimation of c_1 and α. Here instead, to keep things simple, we numerically estimate the minimum of MSE by repeatedly computing $V^{(n)}$. Note that here I only write the algorithm for computing $\mu_{N,M}^{(n)}$, but should be very similar for $\sigma_{N,M}^{(n)}$.

Algorithm 1 A-MFMC

Require: A set of N input-output samples $\boldsymbol{x}_1, f(\boldsymbol{x}_1), \boldsymbol{x}_2, f(\boldsymbol{x}_2) ..., \boldsymbol{x}_N, f(\boldsymbol{x}_N)$, and a set of inputs $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_M$, with $M \gg N$. Assume that training and evaluating $f^{(n)}$ has negligible runtime compared to the cost of running $f(\boldsymbol{x})$.

for $n = 5 \dots N$ **do**
  Split the first n samples into n_{train} and n_{val} with an $80\% - 20\%$ split.
  Fit a Lasso model $f^{(n_{train})}$.
  Estimate $V^{(n_{train})} = \text{Var}\left(f^{(n_{train})} - f\right)$ on the n_{val} samples.
  Compute $\text{MSE}_{n_{train}} = \frac{V^{(n_{train})}}{N - n_{train}}$.
end for
Find $n^\star = \min_{n_{train}} \left(\text{MSE}_{n_{train}}\right)$.
Use $f^{(n^\star)}$ to compute the estimator $\mu_{N,M}^{(n^\star)}$.

1.2.2 Questions

- One disadvantage of normal MFMC is that it might be less accurate than simple MC, especially in high dimensions. With LMC we showed that it is always "equally or more accurate than simple MC", but in fact A-MFMC also has this property (if the surrogate model is very poor e.g. in high dimensions, then the optimal will be $n^\star = 0$ which is exactly equivalent to simple MC).
- Is A-MFMC more, equally, or less accurate than LMC?

- If not, is there any other advantage of using LMC over A-MFMC?