项目作业二：自适应 MC 方法的可行性探索

一、摘要

本项目报告简要阐述了一种基于重要性抽样的自适应 MC 积分方法，此方法的思想来源于文献[2]。报告先简要说明简单的 MC 算法以及方差缩减技术的必要性，然后引出一种常用的方差缩减方法：重要性抽样。之后，我们介绍重要性抽样实施的困难，最后详述一种不依赖于先验信息的重要性抽样方法：自适应重要性抽样。

二、Naïve Monte-Carlo 方法

实际应用中，Monte-Carlo 方法主要用于高效地计算具有如下形式的积分问题：

$$\mu = \mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x)f(x)dx$$

其中 $f(x)$ 为某个随机变量的概率密度函数。最朴素的 Monte-Carlo 方法根据概率密度 $f(x)$ 生成一系列样本 $(X_1, \dots, X_m)$，并计算 $h(x)$ 的样本均值，作为 μ 的估计值：

$$\hat{\mu}^{MC} = \frac{1}{m}\sum_{j=1}^{m} h(x_j)$$

根据强大数定律，$\bar{h}_m \to \mu, a.s.$ 此估计在均方意义下的误差为：

$$\mathrm{Var}(\hat{\mu}^{MC}) = \frac{1}{m}\mathrm{Var}_f[h(X)] = \frac{1}{m}\int_{\mathcal{X}} (h(x)-\mu)^2 f(x)dx$$

因此，为了提高估计的精度，有两个方面可以入手，一是样本的数量，二是方差项 $\mathrm{Var}_f[h(X)]$。方差项的缩小有很多手段，下面介绍的重要性抽样就是其中一种。

三、Importance Sampling 方法

重要性抽样方法基于这样一种思想：一个给定的积分通常可以用多种不同的形式来描述。例如，假设 g(x) 为另一个概率密度函数，其支撑集包含 $f(x)$ 的支撑集：$supp(f) \subset supp(g)$，那么所求积分可以表示为如下形式：

$$\mu = \int_{\mathcal{X}} h(x)\frac{f(x)}{g(x)}g(x)dx = \mathbb{E}_g\left[h(X)\frac{f(X)}{g(X)}\right]$$

因此，我们可以根据概率密度 $g(x)$ 生成样本 $(X_1, \dots, X_m)$，由此得到 μ 的另一个无偏估计量：$\hat{\mu}_g^{IS} = \frac{1}{m}\sum_{j=1}^{m}\frac{f(X_j)}{g(X_j)}h(X_j)$。

这个估计量的均方误差为：

$$\mathrm{Var}\left(\hat{\mu}_g^{IS}\right) = \frac{1}{m}\mathrm{Var}_g\left[h(X)\frac{f(X)}{g(X)}\right] = \frac{1}{m}\int_{\mathcal{X}}\left(h(x)\frac{f(x)}{g(x)}-\mu\right)^2 g(x)dx$$

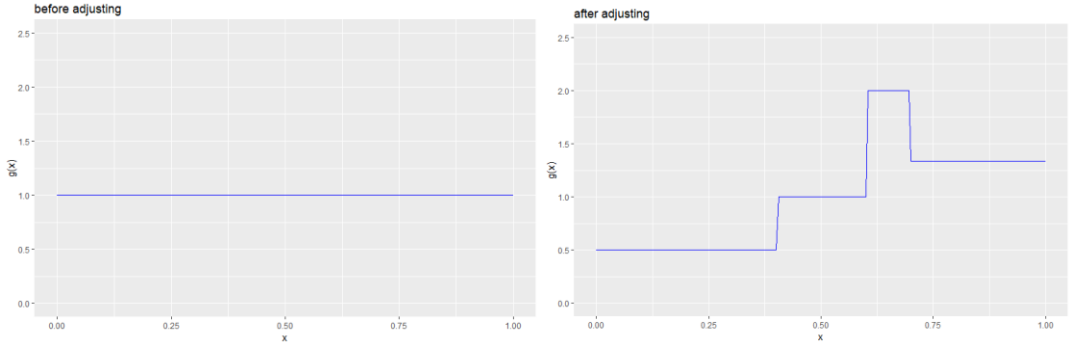
容易看出，$g(x)$ 的选取对上述误差项有很大的影响，所以在重要性分布方法中，$g(x)$ 的选择需要十分谨慎。如果选取的不好，可能还会增大估计的误差。理论上，为使上述误差项达到最小，应选择 $g(x)$ 为 (证明见文献[1] Theorem 3.12.)：

$$g^*(x) = \frac{|h(x)|f(x)}{\int_{\mathcal{X}}|h(z)|f(z)dz}$$

因此，在使用重要性抽样方法时，理论上应选择 $g^*(x)$。不过，由于 $g^*(x)$ 的计算涉及计算另一个积分 $\mathbb{E}_f|h(X)|$，这给此方法的应用带来了困难，所以实际应用中只能选择接近于 $g^*(x)$ 的分布。这可以通过两种手段来实现，一是利用某些特殊的先验信息，不过这通常难以实现。另一种就是利用生成的样本来逼近 $g^*(x)$ 的分布，这就是下面介绍的自适应重要性抽样方法。

四、 自适应 Importance Sampling 方法

假设我们需要估计积分：$I=\int_0^1 h(x)dx$。那么可以使 $f(x)$ 以及初始的 $g_0(x)$ 都为均匀分布的密度函数，然后通过生成的样本来获取 $\mathbb{E}_f|h(X)|$ 的信息，以此不断迭代并调整 $g(x)$，使其越来越接近于 $g^*(x)$。这个方法来源于 G. Peter Lepage 于 1978 年的论文[2]。在调整分布的过程中，一个重要的考虑是不能产生形式太复杂的分布，这样方差减少带来的利好会远不及生成随机样本时付出的代价。因此，一个简单的选择是，每次更新都生成一个阶梯分布函数。如下所示：



左图为初始的 $g(x)$，右图为更新之后的 $g(x)$。另外，为了减少生成样本的负担，以及避免迭代到后期阶梯函数的跳跃点太多，我们在更新 $g(x)$ 时做如下约束：

1. 阶梯函数中跳跃点数量固定，设为 N。
2. 阶梯函数中样本出现于 N 段中的每一段的概率相等。

根据这两个约定，我们就可以通过以下方法产生分布为 $g(x)$ 的样本：产生一个 1 到 N 之间的随机整数，代表其落入的区间段，再产生一个 $[0,1]$ 之间的随机数 (均匀分布)，然后将其线性映射到该区间内。并且，$g(x)$ 可以由分割点 $0=p_0\le p_1\le\cdots\le p_N=1$ 唯一描述。假设在某一次迭代中，产生的样本 $(X_1,\dots,X_m)$ 具有密度 $g(x)$，那么根据此样本可以得到估计 $\hat{I}$ 和样本方差 $\hat{\sigma}^2$。然后为了进行下一步迭代，需要按照以下策略更新 $g(x)$：

1. 根据样本 $(X_1,\dots,X_m)$ 统计 $|h(x)|$ 的分布情况：

$$h_i\equiv\sum_{p_{i-1}<x\le p_i}|h(x)|\ \propto\ \frac{1}{p_i-p_{i-1}}\int_{p_{i-1}}^{p_i}|h(x)|dx$$

2. 利用 h_i 将原先的每个区间 $[p_{i-1},p_i]$ 等距划分为 m_i 个小区间，其中

$$m_i=K\frac{h_i(p_i-p_{i-1})}{\sum_{j=1}^N h_j(p_j-p_{j-1})}$$

K 为预先选定的常数，通常为 1000 左右。容易看出，当区间中 $|h(x)|$ 的数值较大时，m_i 也会较大。

3. 为了保持区间数量不变，对第 2 步中划分得到的小区间，每 $(\sum_{i=1}^N m_i)/N$ 个合并为一个新的大区间。这样得到新的划分点 $0=\tilde{p}_0\le\tilde{p}_1\le\cdots\le\tilde{p}_N=1$ 以及相应的 $\tilde{g}(x)$。

根据上述方法进行 t 次迭代之后，最后得到的估计量以及误差为：

$$\hat{\mu}^{AIS}=(\hat{\sigma}^{AIS})^2\sum_{k=1}^t\frac{\hat{I}_k}{\hat{\sigma}_k^2},\qquad \hat{\sigma}^{AIS}=\left(\sum_{k=1}^t\frac{1}{\hat{\sigma}_k^2}\right)^{-\frac{1}{2}}$$

其中 $\hat{I}_k$ 和 $\hat{\sigma}_k^2$ 分别为第 k 次迭代得到的积分估计值以及样本方差。这个估计量的无偏性以及相合性容易证明。

报告中只讨论了一维的情况，至于高维的情形，此方法也有自然的推广，详见文献[2]。

[1] Robert, Christian & Casella, George. (2000). Monte Carlo Statistical Method. Technometrics. 42. 10.2307/1270959.
[2] Peter Lepage, G. (1978). A New Algorithm for Adaptive Multidimensional Integration. Journal of Computational Physics. 27. 192-203. 10.1016/0021-9991(78)90004-9.
[3] Neufeld, James. (2015). Adaptive Monte Carlo Integration. 10.13140/RG.2.2.31929.16483.