Clustered Multilevel Monte Carlo (C-MLMC)

May 31, 2025

1 Problem Setup and Notation

We aim to estimate $\mathbb{E}[P]$ for a path-dependent payoff P, evaluated under an SDE model. Using the Multilevel Monte Carlo (MLMC) framework, we define estimators:

$$\Delta P_\ell = \begin{cases} P_0, & \text{if } \ell = 0, \\ P_\ell - P_{\ell-1}, & \text{if } \ell \geq 1. \end{cases}$$

Let $V_\ell = \mathrm{Var}[\Delta P_\ell]$, and C_ℓ be the average cost per sample on level ℓ. Classical MLMC chooses

$$N_\ell \propto \sqrt{\frac{V_\ell}{C_\ell}}$$

to achieve RMSE ε, with:

$$\sum_{\ell=0}^{L} \frac{V_\ell}{N_\ell} \leq (1-\theta)\varepsilon^2.$$

2 Motivation for Clustering

When the variance is highly concentrated in rare regions of the input space (e.g., digital or exotic options), naive sampling wastes budget on low-variance samples. The C-MLMC approach improves efficiency by:

- Extracting features from simulation metadata,
- Clustering samples to identify variance modes,
- Estimating per-cluster variance $V_{\ell,c}$ and probabilities $P_{\ell,c}$,
- Allocating samples in proportion to $\sqrt{P_{\ell,c} V_{\ell,c}}$.

3 C-MLMC Algorithm

Algorithm 1 Clustered MLMC Estimator

Require: Simulator $f(\ell, N, \texttt{return\_details})$, feature extractor φ, tolerance ε, pilot size N_0, clusters n_c

Initialize $L \leftarrow L_{\min}$, $\theta \in (0,1)$
for each level $\ell \leq L$ **do**
  Run pilot: $\{(Y_i, d_i)\} \leftarrow f(\ell, N_0, \texttt{true})$
  Compute features $\phi_i = \varphi(d_i)$
  Cluster features using K-means into n_c clusters
  Estimate cluster variances $V_{\ell,c}$ and probabilities $P_{\ell,c}$
end for
while not converged **do**
  Compute level variances $V_\ell = \sum_c P_{\ell,c} V_{\ell,c}$
  Estimate level costs C_ℓ
  Allocate samples $N_\ell \propto \sqrt{V_\ell / C_\ell}$
  Allocate cluster-wise $N_{\ell,c} \propto \sqrt{P_{\ell,c} V_{\ell,c}}$
  for each cluster c **do**
    Sample until $N_{\ell,c}$ accepted samples with cluster label c
    Accumulate $\sum Y$, $\sum Y^2$, update cost
  end for
  Bias check via extrapolation; if too large, increment L
end while
Output estimate $\hat{P} = \sum_{\ell=0}^{L} \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} Y_{\ell,i}$

4 Why $\sqrt{P_c V_c}$ Allocation?

To minimize estimator variance:

$$\mathrm{Var}(\hat{Y}) = \sum_c \frac{P_c^2 V_c}{N_c} \quad \text{subject to} \quad \sum_c N_c = N.$$

Using Lagrange multipliers, the optimal allocation satisfies:

$$N_c \propto P_c \sqrt{V_c} \quad \Rightarrow \quad \text{weights} \propto \sqrt{P_c V_c}.$$

5 Unbiasedness

Each level estimator:

$$\hat{Y}_\ell = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} Y_{\ell,i}$$

is an unbiased estimate of $\mathbb{E}[\Delta P_\ell]$, since stratified sampling preserves marginal distributions. The full estimator:

$$\hat{P} = \sum_{\ell=0}^{L} \hat{Y}_\ell$$

is unbiased for $\mathbb{E}[P_L]$, with bias $\mathbb{E}[P - P_L]$ controlled via extrapolation.

6 Conclusion

C-MLMC enhances classical MLMC by identifying and focusing computation on variance-dominant clusters. The allocation $\propto \sqrt{P_c V_c}$ provides a provably efficient strategy to reduce variance while retaining unbiasedness and convergence guarantees. The result is a more robust and scalable Monte Carlo estimator for path-dependent problems.