Clustered Multilevel Monte Carlo (C-MLMC)

May 16, 2025

1 Problem Setup and Notation

We wish to estimate $\mathbb{E}[P]$ under an SDE via MLMC. Let

$$P_\ell = P\big(S_T^{(\ell)}\big), \quad \Delta P_\ell = \begin{cases} P_0, & \ell = 0, \\ P_\ell - P_{\ell-1}, & \ell \geq 1. \end{cases}$$

Classic MLMC chooses numbers N_ℓ of samples at each level ℓ to control the RMS error ε, using

$$V_\ell = \mathrm{Var}\big[\Delta P_\ell\big], \qquad C_\ell = \text{cost per sample at level } \ell,$$

and allocates

$$N_\ell \;\propto\; \sqrt{\frac{V_\ell}{C_\ell}} \quad \text{subject to} \quad \sum_{\ell=0}^{L} \frac{V_\ell}{N_\ell} \leq (1-\theta)\,\varepsilon^2.$$

2 Added Value: Clustering for Variance Reduction

In the digital-option example, most ΔP_ℓ are zero except for rare "boundary" paths. Blind MLMC wastes effort on the sea of zeros. Our C-MLMC enhances MLMC by:

- **Pilot sampling** to learn where ΔP_ℓ is nonzero.
- **Feature extraction** from each path's `details` (e.g. distance to strike).
- **K-means clustering** of these features into n_c clusters.
- **Per-cluster variance** estimates $V_{\ell,c}$ to identify "high-impact" clusters.
- **Stratified allocation** of the usual MLMC extra samples dN_ℓ across clusters.

3 C-MLMC Algorithm

Algorithm 1 Clustered MLMC Estimator

Require: level function $f(\ell, N, \texttt{return\_details})$, feature map φ, tolerance ε, pilot size N_0, clusters n_c
Ensure: estimate $\hat{P}$

Initialize $L \leftarrow L_{\min},\ \theta \in (0,1)$
Initialize arrays $N_\ell,\ S_\ell,\ C_\ell,\ V_{\ell,c}$ for $\ell = 0, \ldots, L$
while not converged **do**
    for $\ell = 0$ **to** L **do**
        if cluster info at ℓ not yet computed **then**
            Run pilot: $(Y_i,\ d_i)_{i=1}^{N_0} \leftarrow f(\ell, N_0, \texttt{true})$
            Compute features $\mathbf{f}_i = \varphi(d_i)$
            Fit K-means on $\{\mathbf{f}_i\} \rightarrow$ labels $\{c_i\}$
            for $c = 1$ **to** n_c **do**
                $V_{\ell,c} \leftarrow \mathrm{Var}(\{Y_i : c_i = c\})$
            end for
        end if
    end for
    Compute total variances $V_\ell = \sum_c V_{\ell,c}$
    Compute costs C_ℓ and allocation $N_\ell \propto \sqrt{V_\ell / C_\ell}$
    Compute extra samples $dN_\ell = N_\ell - \mathrm{used}_\ell$
    for $\ell = 0$ **to** L **do**
        if $dN_\ell > 0$ **then**
            Split $dN_\ell \rightarrow \{dN_{\ell,c}\}$ proportional to $V_{\ell,c}$
            for $c = 1$ **to** n_c **do**
                Draw cluster-wise: simulate until $dN_{\ell,c}$ samples in cluster c
                Accumulate sums and costs
            end for
        end if
    end for
    Check bias remainder; if too large, increment L and reinitialize for new level
end while
$\hat{P} \leftarrow \sum_{\ell=0}^{L} \frac{S_\ell}{N_\ell}$
return $\hat{P}$

4 Why Clustering Helps

- **Detects Rare Events:** Clusters with large $V_{\ell,c}$ correspond to paths near payoff discontinuities.
- **Focuses Effort:** Stratification directs samples to high-impact clusters, reducing overall variance.
- **Preserves MLMC Guarantees:** We still meet both bias and variance tolerances.