Monte Carlo simulation algorithm

June 2024

Monte Carlo methods represent a broad category of computational algorithms employed in optimization, numerical integration, and sample generation from probability distributions, a process known as Monte Carlo simulation. The central idea of these methods is to use randomness to solve problems that could be deterministic. The fundamental principle of Monte Carlo simulation is that the behavior of a statistic can be evaluated through an empirical process, obtained by generating and analyzing several random samples [1].

The computational cost associated with Monte Carlo simulation can be considerably high. Generally, the method requires a large number of samples to achieve a good approximation, which can result in significant execution time. In summary, the method relies on repeated random sampling to obtain useful numerical results, especially in situations where other approaches are difficult or unfeasible.

The estimation of $p(z)$ is implicitly obtained through Monte Carlo sampling that uses the estimated values of $\boldsymbol{\theta}^{(t)}$, $\widehat{\mathbf{P}}^{(\mathbf{t})}(\mathbf{0})$, or $\widehat{\mathbf{P}}^{(\mathbf{t})}(\mathbf{1})$. Specifically, we generate a sample of $p(z)$ by traversing the schedule day by day and drawing Bernoulli random variables with parameters $\widehat{\theta}_j^{(t)}$, $\widehat{P}_j^{(t)}(0)$, or $\widehat{P}_j^{(t)}(1)$ to determine if task t can be performed on day j of the year at the project site. We track the tasks that could not be performed and adjust the schedule accordingly. Indexing the simulations as $b = 1, \ldots, B$, where Z_b is, by definition, a sample of $p(z)$, representing the time required for project completion in the given simulation. Algorithm 1 details our procedure for a set of tasks with finish-to-start dependencies, but other dependencies can be easily incorporated into this framework.

The Monte Carlo sample $Z_1, \ldots, Z_B$ can be used to estimate $p(z)$ and calculate its statistics. One possible approach is to apply a kernel density estimator to $Z_1, \ldots, Z_B$ to approximate Z by a continuous distribution and use a smoother pattern [2]. Alternatively, discrete versions can be chosen [3]. Furthermore, the α-quantile of $p(z)$ can be approximated by calculating the α-quantile of the samples. Similarly, the mean of Z, given by $\mathbb{E}[Z] = \sum_z z p(z)$, can be approximated by $B^{-1} \sum_{b=1}^{B} Z_b$.

References

[1] Christopher Z Mooney. *Monte carlo simulation*. 116. Sage, 1997.

[2] EA Nadaraya. “On non-parametric estimates of density functions and regression curves”. In: *Theory of Probability & Its Applications* 10.1 (1965), pp. 186–190.

Algorithm 1 Project Execution Time Estimation

Require: set of tasks $\mathcal{T}$, project start date $start\_date$, duration d_t of each task t, execution probabilities $\widehat{\boldsymbol{\theta}}^{(t)}$, or $\widehat{\mathbf{P}}^{(\mathbf{t})}(\mathbf{0})$ and $\widehat{\mathbf{P}}^{(\mathbf{t})}(\mathbf{1})$, number of Monte Carlo samples B

Ensure: Set of samples Z_b, representing the number of days taken to complete the project in the b-th Monte Carlo sample

for b in $\{1, \ldots, B\}$ **do**
    while the set $\mathcal{T}$ is not finished **do**
        for t in $\mathcal{T}$ that can start **do**
            $start\_date_t \leftarrow$ start date of task t
            $successes \leftarrow 0$
            while $successes < d_t$ **do**
                $j \leftarrow$ day of the year of $start\_date_t$
                if $mc = True$ **then**
                    Draw $X \sim \text{Ber}(P_j^{(t)}(X_{-1}))$
                else
                    Draw $X \sim \text{Ber}(\widehat{\theta}_j^{(t)})$
                end if
                if $X = 1$ **then**
                    $successes \leftarrow successes + 1$
                end if
                if $successes < d_t$ **then**
                    $start\_date_t \leftarrow start\_date_t + 1$ day
                end if
                if $mc = True$ **then**
                    $X_{-1} \leftarrow X$
                end if
            end while
        end for
    end while
    $end\_date \leftarrow max\{start\_date_t$ for each task t in $\mathcal{T}\}$
    $Z_b \leftarrow end\_date - start\_date$
end for
return Z_b =0

[3] Balaji Rajagopalan and Upmanu Lall. “A kernel estimator for discrete distributions”. In: *Journaltitle of Nonparametric Statistics* 4.4 (1995), pp. 409–426.