The Stochastic Grid Bundling Method

Ricardo Rincón

March 4, 2020

Premia 22
Closed Formula Methods

Shashi Jain and Cornelis W. Oosterlee (2013) [**?**] proposed the Stochastic Grid Bundling Method (SGBM) for pricing of Bermudan options with several underlying assets. The method is based on simulation, regression, and bundling. The steps involved in the SGBM algorithm, are:

- Step I: Generating grid points
- Step II: Option value at terminal time
- Step III: Bundling
- Step IV: Mapping high-dimensional state space to a low-dimensional space
- Step V: Computing the continuation and option values at t_{m-1}

We perform initially the first two steps and then, starting from t_M and moving backwards in time, steps III to V are performed for each time step, t_m, $m \leq M$. The next sections retake the method exposed in [**?**], for more details and the proofs of all theoretical results one can look into the paper.

Initial steps

We start by doing a Monte Carlo simulation, following the indicated scheme for the required financial model, of the stochastic paths. Simulating independent copies of the market state variable, $\{\mathbf{S}_{t_0}(n), ..., \mathbf{S}_{t_M}(n)\}$, $n = 1, ..., N$, is a forward process in which all the paths start from the same initial state $\mathbf{S}_{t_0}$.

Subsequently the option value for all paths at terminal time is computed as :

$$V_{t_M}(\mathbf{S}_{t_M}) = \max\left(h(\mathbf{S}_{t_M}), 0\right).$$

Bundling

We want to sample the distribution of $\mathbf{S}_{t_m}$ conditional on the state $\mathbf{S}_{t_{m-1}}$, in a backward loop. To accomplish this, the SGBM firstly clusters using some measure of proximity the grid points at t_{m-1} into ν non-overlapping

partitions. Then, we sample $\mathbf{S}_{t_m}$ with those paths originated from the bundle that contains $\mathbf{S}_{t_{m-1}}$. We have considered the three different approaches for partitioning proposed by Jain and Oosterlee (2013) [**?**] for SGBM and we explain them next.

K-means clustering algorithm

The objective of this algorithm is to cluster points so as to minimize the sum of squares within clusters, i.e :

$$\arg\min_{\mathcal{B}_{t_{m-1}}} \sum_{\beta=1}^{\nu} \left(\sum_{\mathbf{S}_{t_{m-1}}(n) \in \mathcal{B}_{t_{m-1}}(\beta)} \|\mathbf{S}_{t_{m-1}}(n) - \mu_\beta\|^2 \right),$$

where μ_β is the mean of the points into each of the ν non-overlapping bundles β, $\mathcal{B}_{t_{m-1}}(\beta)$. The algorithm (Lloyd 1982 [2]) uses an iterative refinement technique. Initially, we take ν aleatory points from the N points to cluster as bundle centroids, $\mu_1^1, ..., \mu_\nu^1$. Subsequently, we perform the following steps alternately :

- **Step 1** : Assign grid points to the set whose mean is closest to it.

  $$\mathcal{B}_{t_{m-1}}^{(l)}(\beta) = \{\mathbf{S}_{t_{m-1}}(n) : \|\mathbf{S}_{t_{m-1}}(n) - \mu_\beta^{(l)}\|^2 \leq \|\mathbf{S}_{t_{m-1}}(n) - \mu_j^{(l)}\|^2, \forall\, 1 \leq j \leq \nu\},$$

  where grid point $\mathbf{S}_{t_{m-1}}(n)$ is assigned to just one bundle.

- **Step 2** : If the assignment of the grid points does not change anymore from a previous iteration the process has converged, else the means are updated into each of the new clusters as :

  $$\mu_\beta^{(l+1)} = \frac{1}{|\mathcal{B}_{t_{m-1}}^{(l)}(\beta)|} \sum_{\mathbf{S}_{t_{m-1}}(n) \in \mathcal{B}_{t_{m-1}}^{(l)}(\beta)} \mathbf{S}_{t_{m-1}}(n),$$

  where $|\mathcal{B}_{t_{m-1}}^{(l)}(\beta)|$ is the cardinal of the set $\mathcal{B}_{t_{m-1}}^{(l)}(\beta)$.

This is computationally the most expensive algorithm of the three presented here. In order to not prejudice *k-means clustering* with respect to the others, especially in high dimensions, we can set a maximum number of iterations before stopping performing those two steps, without having converged. Even in this case, we find an accurate price. In addition, we can specify directly centroids and just distribute grid points in clusters as indicated in step 1. We use this procedure when computing path estimator with centroids previously found in calibration phase.

Recursive bifurcation

To bundle grid points we can also perform the following steps :

- **Step 1** : Compute the mean of the grid points along each dimension,

$$\mu_\delta = \frac{1}{N}\sum_{n=1}^{N} S^\delta_{t_{m-1}}(n),\ \delta = 1, ..., d.$$

- **Step 2** : Bundle separately along each dimension the grid points by dividing the grid into 2^d sets according to :

$$A_\delta = \{\mathbf{S}_{t_{m-1}}(n) : S^\delta_{t_{m-1}}(n) > \mu_\delta,\ n = 1, ..., N\},$$

$$\bar{A}_\delta = \{\mathbf{S}_{t_{m-1}}(n) : S^\delta_{t_{m-1}}(n) \leq \mu_\delta,\ n = 1, ..., N\},$$

  where $\delta = 1, ..., d$.

- **Step 3** : The 2^d unique non-overlapping clusters are obtained intersecting these sets as follows :

$$\begin{aligned}
\mathcal{B}_{t_{m-1}}(1) &= A_1 \cap A_2 \cap ... \cap A_d,\\
\mathcal{B}_{t_{m-1}}(2) &= \bar{A}_1 \cap A_2 \cap ... \cap A_d,\\
\mathcal{B}_{t_{m-1}}(3) &= A_1 \cap \bar{A}_2 \cap ... \cap A_d,\\
&\vdots\\
\mathcal{B}_{t_{m-1}}(2^d) &= \bar{A}_1 \cap \bar{A}_2 \cap ... \cap \bar{A}_d,
\end{aligned} \tag{1}$$

We can continue performing as much iterations as we want of previous steps to split further each bundle.

Recursive bifurcation of reduced state space

We can also bundle the grid points based on proximity of the reduced state space $h(\mathbf{S}_{t_{m-1}})$, i.e. starting by using the payoff as a mapping function and then employing the clustering procedure of the *recursive bifurcation* to the mapped points (which belong to dimension $d = 1$). However, in terms of programming, we do not call the *recursive bifurcation* function after obtaining the reduced state space, because of feasible optimizations in the algorithm and memory allocation. The number of bundles obtained after p iterations in this case will be 2^p.

Mapping high-dimensional state space to a low-dimensional space

Corresponding to each bundle $\mathcal{B}_{t_{m-1}}(\beta)$, $\beta = 1, ..., \nu$, a parametrized value function $Z : \mathbb{R}^d \times \mathbb{R}^K \to \mathbb{R}$, is introduced. This approximation of the value function assigns values $Z(\mathbf{S}_{t_m}, \alpha^{\beta}_{t_m})$ to states $\mathbf{S}_{t_m}$ in order to better deal with the large dimension of the state space. $\alpha^{\beta}_{t_m} \in \mathbb{R}^K$ is a vector of free parameters and we aim to choose, for each t_m and bundle β, a parameter vector so that $Z(\mathbf{S}_{t_m}, \alpha^{\beta}_{t_m}) \approx V_{t_m}(\mathbf{S}_{t_m})$.

We use basis functions that map the state space from $\mathbb{R}^d$ to $\mathbb{R}$, to approximate the value functions. The function $Z(\mathbf{S}_{t_m}, \alpha^{\beta}_{t_m})$ projecting the option values onto the span of ϕ is restricted to a linear combination of basis functions and, can be approximated by :

$$Z(\mathbf{S}_{t_m}, \hat{\alpha}^{\beta}_{t_m}) = \sum_{k=1}^{K} \hat{\alpha}^{\beta}_{t_m}(k)\phi_k(\mathbf{S}_{t_m}), \tag{2}$$

satisfying,

$$\arg\min_{\hat{\alpha}^{\beta}_{t_m}} \sum_{n=1}^{|\mathcal{B}_{t_{m-1}}(\beta)|} \left(V_{t_m}(\mathbf{S}_{t_m}(n)) - \sum_{k=1}^{K} \hat{\alpha}^{\beta}_{t_m}(k)\phi_k(\mathbf{S}_{t_m}(n)) \right)^2. \tag{3}$$

Therefore, the parametrized function $Z(\mathbf{S}_{t_m}, \hat{\alpha}^{\beta}_{t_m})$ is computed, corresponding to each bundle $\mathcal{B}_{t_{m-1}}(\beta)$, using ordinary least squares regression, so that:

$$V_{t_m}(\mathbf{S}_{t_m}(n)) = Z(\mathbf{S}_{t_m}(n), \hat{\alpha}^{\beta}_{t_m}) + \epsilon^{\beta}_{t_m}, \tag{4}$$

where $\mathbf{S}_{t_{m-1}}(n) \in \mathcal{B}_{t_{m-1}}(\beta)$ and $\epsilon^{\beta}_{t_m}$ is the error made in the regression. In theoretical terms, it is assumed that $\mathbb{E}[\epsilon^{\beta}_{t_m} | \mathbf{S}_{t_{m-1}}(n)] = 0$.

The continuation and option values at t_{m-1}

Now, using the parametrized option value function $Z(\mathbf{S}_{t_m}, \hat{\alpha}^{\beta}_{t_m})$ corresponding to bundle $\mathcal{B}_{t_{m-1}}(\beta)$, the continuation values for the paths into this cluster are approximated by :

$$\hat{Q}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n)) = D_{t_{m-1}} \mathbb{E}[Z(\mathbf{S}_{t_m}, \hat{\alpha}^{\beta}_{t_m}) | \mathbf{S}_{t_{m-1}} = \mathbf{S}_{t_{m-1}}(n)], \tag{5}$$

where $\mathbf{S}_{t_{m-1}}(n) \in \mathcal{B}_{t_{m-1}}(\beta),\ n = 1, ..., N,\ \beta = 1, ..., \nu$. Using Equation (2), this can be written as:

$$\begin{aligned}\hat{Q}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n)) &= D_{t_{m-1}}\mathbb{E}\left[\left(\sum_{k=1}^{K}\hat{\alpha}_{t_m}^{\beta}(k)\phi_k(\mathbf{S}_{t_m})\right)|\mathbf{S}_{t_{m-1}} = \mathbf{S}_{t_{m-1}}(n)\right] \\ &= D_{t_{m-1}}\sum_{k=1}^{K}\hat{\alpha}_{t_m}^{\beta}(k)\mathbb{E}[\phi_k(\mathbf{S}_{t_m})|\mathbf{S}_{t_{m-1}} = \mathbf{S}_{t_{m-1}}(n)].\end{aligned} \tag{6}$$

The **direct estimator** of the option values for the paths at t_{m-1} is defined as :

$$\hat{V}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n)) = \max(h(\mathbf{S}_{t_{m-1}}(n)),\ \hat{Q}_{t_{m-1}}(\mathbf{S}_{t_{m-1}}(n))),$$

where $n = 1,\ \ldots,\ N$. The direct estimator is said to be an upper bound, $\mathbb{E}[\hat{V}_{t_0}(\mathbf{S}_{t_0})] \geq V_{t_0}(\mathbf{S}_{t_0})$, converging to the true price when simulating an increasing number of paths and using an increasing number of bundles to cluster paths at each time step.

Numerically, if the regression is not perfectly performed, we will not obtain $\mathbb{E}[\epsilon_{t_m}^{\beta}|\mathbf{S}_{t_{m-1}}(n)] = 0$. In addition, to compute $\mathbb{E}[\phi_k(\mathbf{S}_{t_m})|\mathbf{S}_{t_{m-1}} = \mathbf{S}_{t_{m-1}}(n)]$, we do not take a closed form or an analytic approximation for each payoff. We use directly the first right hand side in (6). Therefore, it is not strange that here, direct estimator is not sometimes an upper bound as it is shown in [**?**] experiments.

Computing path estimator

After finishing previous procedure to compute the direct estimator, we simulate a new set of paths and we develop a lower bound estimator based on these new paths. Using the same scheme followed in prior generation of grid points, we simulate $\mathbf{S}(n) = \mathbf{S}_{t_1}(n),\ \ldots,\ \mathbf{S}_{t_M}(n), n = 1,\ \ldots,\ N_L$. Along each path, the approximate optimal policy exercises at,

$$\hat{\tau}^*(\mathbf{S}(n)) = \min\{t_m : h(\mathbf{S}_{t_m}(n)) \geq \hat{Q}_{t_m}(\mathbf{S}_{t_m}(n)), m = 1,\ \ldots,\ M\},$$

where $\hat{Q}_{t_m}(\mathbf{S}_{t_m}(n))$ is computed using Equation (6), which means to compute another backward loop including bundling again, but using directly the parameters $\alpha_{t_m}^{\beta}$ stored before for the regression step.

The **path estimator** $\underline{V}_{t_0}(\mathbf{S}_{t_0})$, lower bound respect to the true option value, is :

$$\underline{V}_{t_0}(\mathbf{S}_{t_0}) = \lim_{N_L}\frac{1}{N_L}\sum_{n=1}^{N_L}h(\mathbf{S}_{\hat{\tau}^*(\mathbf{S}(n))}) \leq V_{t_0}(\mathbf{S}_{t_0}).$$

The SGBM: Calculation of Exposure Profiles and CVA

Feng and Oosterlee (2014) [1] proposed using the Stochastic Grid Bundling Method (SGBM) for computation of exposure profiles. For more details and a theoretical framework one can see [1].

The holder receives the payoff value, $g(S_m)$, when the option is exercised. When the option contract is still alive ($t_m < \tau_m$), the discounted option value, the continuation value w.r.t. state vector $\mathbf{X}_m$, is

$$\hat{Q}_m(\mathbf{X}_m) := \mathbb{E}^{\mathbb{Q}} \left[V_{m+1}(\mathbf{X}_{m+1}) \cdot D(t_m, t_{m+1}) | X_m \right], \tag{7}$$

where $V_{m+1}(\cdot)$ represents the option value at time t_{m+1}.

For European options, denoting S_m as the underlying asset variable at time t_m, the option value equals the continuation value before maturity and the holder receives the payoff value only at maturity, i.e.

$$V_m^{Euro}(\mathbf{X}_m) = \begin{cases} g(S_M), & \text{for } t_M, \\ \hat{Q}_m(\mathbf{X}_m), & \text{for } t_m \in \mathcal{T} - t_M. \end{cases} \tag{8}$$

For Bermudan options, we assume that the credit information of the other party does not influence the exercise decision of the option holder. At each exercise date the holder compares the payoff value with the continuation value of the option, based on the currently available information. The holder keeps the option until the payoff value is higher. When the option is still alive at time t_m, denoting $\mathcal{T}_e$ the exercise dates, the option can be computed as :

$$V_m^{Berm}(\mathbf{X}_m) = \begin{cases} \max\{\hat{Q}_m(\mathbf{X}_m), g(S_m)\}, & \text{for } t_m \in \mathcal{T}_e, \\ \hat{Q}_m(\mathbf{X}_m), & \text{for } t_m \in \mathcal{T} - \mathcal{T}_e. \end{cases} \tag{9}$$

On the other hand, the exposure value becomes 0 when the option is exercised as there is not possible economic loss for the contract holder any longer,

$E_M = 0$. Equally, for Bermudan options, after being exercised at time t_m the exposure later is 0. By definition, the value of the exposure can be represented mathematically as :

$$E_m(\mathbf{X}_m) = \begin{cases} 0, & \text{when the option is exercised,} \\ V_m(\mathbf{X}_m), & \text{when the option is alive.} \end{cases} \tag{10}$$

where $E_m(\cdot)$ represents the exposure at time t_m, $m = 1, 2, \ldots, M-1$.

Having computed the exposure values for all the simulated paths at times t_m, $m = 0, \ldots, M-1$, the EE value at time t_m is approximated as an average of them :

$$EE(t_m) \approx \frac{1}{N} \sum_{i=1}^{N} E_m(\hat{\mathbf{x}}_m(i)), \tag{11}$$

where N represents the number of paths and $\hat{\mathbf{x}}_m(i)$, $i = 1, \ldots, N$, the values of the state variables of the i-th path at t_m. Since the interest rate is deterministic in our case, the discounted exposure (EE^*) is the product of the discount factor and the precedent EE value and so we also can obtain the CVA, the direct estimator CVA. A discrete version formula to compute the CVA can be given as :

$$CVA \approx (1-\delta) \sum_{m=0}^{M-1} EE^*(t_m)(PD(t_{m+1}) - PD(t_m)). \tag{12}$$

where δ is the recovery rate and PD(s) is the default probability function, $PD(t) = 1 - \exp\left(-\int_0^t h(t)dt\right)$, with $h(t)$ called the intensity.

Backward iteration for exposure values

We provide here the procedure for calculating the exposure values in a backward iteration, starting at final time T. This procedure corresponds to the one introduced in [1] as well.

At time t_M, as we mentioned before, the exposure values are 0 as there is not possible economic loss for the holder any longer. For each path, the option value is calculated as $V_M(\hat{\mathbf{x}}_M(i))$, with the corresponding formulas of European or Bermudan options, (8) and (9).

At time t_{M-1}, for all the paths, the continuation values $\hat{Q}_{M-1}(\hat{\mathbf{x}}_{M-1})$, $i = 1, \ldots, N$, can be calculated with (7), and so the option values, $V_{M-1}(\hat{\mathbf{x}}_M(i))$, (8) or (9), and the exposure values, $E_{M-1}(\hat{\mathbf{x}}_M(i))$, (10), can be also computed.

Then the iteration goes backward in time repeating the bundling and regression exposed in SGBM explication to compute the continuation, option, and exposure values at each time step until we arrive to the initial time. When we arrive at t_0, we have the option and exposure values of every grid point.

For Bermudan options, we also need to take into account the optimal early-exercise strategy in this case. If the option is still alive at t_m, both option and exposure values are set to the corresponding continuation value. Then, the payoff value is calculated for each path, and compared with the continuation value to determine if the option should be exercised. If yes, the exposure at this path from time t_m will be 0, and the option value at time t_m corresponds to the payoff value. The EE function is then written as :

$$EE(t_m) \approx \frac{1}{N} \sum_{\tau_i > t_m} (D(t_m, \tau_i) \cdot \text{cash-flow}(i)), \tag{13}$$

where τ_i is the exercise time of the i-th path, and the cash-flow is the payoff value at time τ_i, $g(S_{\tau_i}(i))$, with $S_{\tau_i}(i)$ the value of the stock of the i-th path at time τ_i. The EE values calculated with precedent expression, provide the path estimator CVA, which considers the obtained "optimal" exercise strategy. For computing the path estimator CVA, we simulate a new set of paths and we employ again, in a backward procedure, the same regression coefficients for each bundle that we used to compute direct estimator (and that we have previously saved at each time step).

Bibliography

[1] Qian Feng and Cornelis W. Oosterlee. Monte carlo calculation of exposure profiles and greeks for bermudan and barrier options under the heston hull-white model. *Available at SSRN 2494233*, 2014. 6, 7

[2] Stuart P. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982. 2