Some basic knowledge

20. April 2019

Let $\mathcal{T}_{tT}$ denotes the set of all stopping time in range $[t,T]$. (Essentially the exercise time of American option)

Definition 0.1. Define the **Buyer's price** as the following:

$$\mathcal{B}_t(F) = \sup\{z \in \mathcal{F}_t \mid \exists(\tau \in \mathcal{T}_{tT}, \Delta) \text{ such that } \pi_B(z,\tau) \geq 0 \ \mathbb{P}^{hist}\text{-a.s.}\} \tag{1}$$

with $\pi_B(z,t)$ the intrinsic value of our portfolio:

$$\pi_B(z,\tau) = -D_{0t}z + \int_t^\tau \Delta_s \cdot d\tilde{X}_s + D_{0\tau}F_\tau \geq 0 \tag{2}$$

Here z is the money account you can otherwise gained from saving; Δ is your hedging position; and $F_\tau = F_\tau(X_t^i : 0 \leq t \leq \tau)$ is the payoff function. Similarly we define the **Seller's price** as:

$$\mathcal{S}_t(F) = \inf\{z \in \mathcal{F}_t \mid \exists\Delta \text{ s.t. } \forall\tau \in \mathcal{T}_{tT}, \text{ such that } \pi_S(z,\tau) := -\pi_B(z,\tau) \geq 0 \ \mathbb{P}^{hist}\text{-a.s.}\} \tag{3}$$

Attention the difference in quantifier. The buyer hat the right to exercise, the seller not.

Remark 0.1. From now onwards I might skip (with or without purpose) some a.s.-arguments just to keep the picture clear.

Theorem 0.2. *Assume there exists an equivalent local martingale measure(ELMM = risk-free measure) $\mathbb{Q} \sim \mathbb{P}^{hist}$. Then:*

$$\mathcal{B}_t(F) \leq \sup_{\tau\in\mathcal{T}_{tT}} \mathbb{E}^{\mathbb{Q}}[D_{t\tau}F_\tau \mid \mathcal{F}_t] = D_{0t}^{-1}\mathcal{S}_t \leq \mathcal{S}_t(F) \tag{4}$$

In particular, in complete market where $\exists!$ ELMM, all three terms are equal.

1 Finite-difference scheme

Now given payoff $g(x)$, our payoff function is $u(t,x) = \sup_{\tau\in\mathcal{T}_{tT}} \mathbb{E}^{\mathbb{Q}}[g(X_\tau) \mid X_t = x]$. The respective semilinear PDE will be:

$$\partial_t u + \mathcal{L}u + (\mathcal{L}g(x))^- \mathbf{1}_{\{g(x)=u(t,x)\}} = 0 \qquad u(t,x) = g(x) \tag{5}$$

with $\mathcal{L}$ our Itô generator: $\mathcal{L} = \sum_i b_i\partial_i + \frac{1}{2}\sum_{1\leq i,j\leq n}\sum_{1\leq k\leq d}\sigma_{i,k}\sigma_{j,k}\partial_{ij}$ and f^- the negative part of function. In the case we take non-zero interest rate, the above becomes:

$$\partial_t u + \mathcal{L}u - ru + (\mathcal{L}g(x) - r(t,x)g(x))^- \mathbf{1}_{\{g(x)=u(t,x)\}} = 0 \tag{6}$$

and dividends, repo etc. can be added in a similar fashion.

Example 1.1. In the case we have American call option on only one dividend-paying asset, the fomrula is early exercise premium formula:

$$\begin{aligned} u(t,X_t) &:= \sup_{\tau\in\mathcal{T}_{tT}} \mathbb{E}^{\mathbb{Q}}[e^{-r(\tau-t)}(X_\tau - K)^+ \mid \mathcal{F}_t] \\ &= \mathbb{E}^{\mathbb{Q}}[e^{-r(T-t)}(X_T - K)^+ \mid \mathcal{F}_t] + \int_t^T e^{-r(s-t)}\mathbb{E}^{\mathbb{Q}}[(rK - qX_s)^- \mathbf{1}_{u(s,X_s)=(X_s-K)^+} \mid \mathcal{F}_t]ds \end{aligned} \tag{7}$$

Note the first term is the fair price of European option, the second term is just a fancy way to describe how much the price would worth should we exercise earlier. I black-boxed all the HJB-principle to derive this.

This will be our testing benchmark. Finite-difference scheme applied to the semilinear PDE is easier to implement, so we can test the accuracy of our Monte-Carlo methods. I sketch it as follow:

- For discrete time this behaves just like European option. We solve it backwards: Given exercise times $t_1 < \cdots < t_N = T$, we first consider solve the original PDE $\partial_t u + \mathcal{L}u - ru = 0$ on $(t_{N-1}, t_N]$ to yield the solution $u(t_{N-1}, x)$. This will serve our boundary condition in the next phase, u.s.w.
- In the continuous case we do the same thing, namely we compute $u(t-dt, x)$ for our timestep dt, and then compare with $g(x)$, and discard the smaller value, and go onwards. The reason this is indeed the approximate solution of the semilinear PDE we mentioned above is actually related to variational inequality. So I will skip here.

2 Monte-Carlo Scheme

First our theorem as discussed earlier:

Theorem 2.1. *Given $\mathcal{M}_{t,0}$ the set of all right-cont martingales $(M_s)_{t\leq s\leq T}$ with $M_t = 0.$, we have:*

$$\sup_{\tau\in\mathcal{T}_{tT}} \mathbb{E}^{\mathbb{Q}}[D_{t\tau}F_\tau \mid \mathcal{F}_t] = \inf_{M\in\mathcal{M}_{t,0}} \left[\sup_{t\leq s\leq T}(D_{ts}F_s - M_s)\middle|\mathcal{F}_t\right] \tag{8}$$

Again keeps in mind that M can be seen as a hedging strategy. So the theorem essentially saying by investing the buyer's price, the seller can hedge in such a way as to cover the payoff even if the buyer exercise optimally.

Remark 2.1 (Discrete Exercise)**.** If we take V_{t_i} to be the fair value of our option, with F_{t_i} the payoff price at time t_i, then the following is intuitive:

$$\begin{cases} V_{t_N} = F_{t_N} \\ V_{t_i} = \max(F_t, \mathbb{E}^{\mathbb{Q}}[D_{t_i t_{i+1}} V_t \mid \mathcal{F}_{t_i}]) \end{cases} \tag{9}$$

Note here the price is defined inductively. Of course we shall exercise if it's already undervalued. we define the **continuation value** $C_{t_i} := \mathbb{E}^{\mathbb{Q}}[D_{t_i t_{i+1}} V_t \mid \mathcal{F}_{t_i}]$ satisfies:

$$\begin{cases} C_{t_N} = -\infty \\ C_{t_i} = \mathbb{E}^{\mathbb{Q}}[D_{t_i t_{i+1}} \max(F_{t_{i+1}}, C_{i+1}) \mid \mathcal{F}_{t_i}] \end{cases} \tag{10}$$

So the optimal stopping time is essentially the first exiting time of the zone $\{V_t \geq F_t\}$:

$$\tau^* = \inf\{t_i \mid V_{t_i} = F_{t_i}\} \tag{11}$$

Remark 2.2. Recall F_t is our payoff function. In our testing case, the options holder receives some continuation coupon $C^c_{t_i}$ at time t_i if he decides not to exercise there, and receives an exercise coupon $C^e_{t_i}$ if otherwise. So the payoff function in such case is:

$$F_{t_i} = \sum_{j<i} D_{t_i t_j} C^c_{t_j} + C^e_{t_i} \tag{12}$$

Note in this case the optimal stopping time is:

$$\tau^* = \inf\left\{t_i \middle| V_{t_i} = \sum_{k<j} D_{t_i t_j} C^c_{t_j} + C^e_{t_i}\right\} \tag{13}$$

Now our Monte-Carlo implementation. The dumbest way would be to simulate all V_{t_i} in all possible model scenarios. And the classical Monte-Carlo method do not provide information on the values of the option at future dates. The first approach **Tsitsiklis-Van Roy(TVR) algorithm** is to optimize this by choosing the path we simulated wisely:

We simulate paths until $T = t_N$ first, then approximates on each path ω the value $V_{t_i}(\omega)$ by:

$$\begin{cases} \hat{V}_{t_N} &= F_{t_N} \qquad \hat{V}_{t_i} = \max(F_{t_i}, \hat{C}_{t_i}) \\ \hat{C}_{t_i} &= \mathbb{E}^{\mathbb{Q}}[D_{t_i t_{i+1}} \hat{V}_{t_{i+1}} \mid \mathcal{F}_{t_i}] \end{cases} \tag{14}$$

namely we follow the scheme of Discrete Exercise. In the original recipe they estimate V_0 directly by $\hat{V}_0 = \mathbb{E}^{\mathbb{Q}}[D_{0t_1}\hat{V}_{t_1}]$, whereas $\mathbb{E}^{\mathbb{Q}}$ is directly estimated by average of all simulated paths.

Remark 2.3 (What is wrong?)**.** This algorithm is not accurate as C_{t_i} is often hard to estimate, as the expectation needs us to estimate the expectation, and even worse the errors builds up as the estimation of $C_{t_{i-1}}$ involves that of C_{t_i}.

In order to reduce this error we instead estimate stopping times: $\tau_i^* = \inf\{t_j \geq t_i \mid C_{t_j} \leq F_{t_j}\}$. Now the scheme **Longstaff-Schwartz(LS)** was built in the following way:

$$\begin{cases} \hat{\tau}_N &= t_N \qquad \hat{\tau}_i = \inf\{t_j \geq t_i \mid \hat{C}_{t_j} \leq F_{t_j}\} \\ \hat{C}_{t_i} &= \mathbb{E}^{\mathbb{Q}}[D_{t_i \hat{\tau}_{i+1}} F_{\hat{\tau}_{t+1}} \mid \mathcal{F}_{t_i}] \end{cases} \tag{15}$$

and we in the end approximate $\hat{V}_0 = \mathbb{E}^{\mathbb{Q}}[D_{0\hat{\tau}}F_{\hat{\tau}}]$, where the expectation is replaced by the average over all empirical paths.

Remark 2.4 (Why better?)**.** Now this is significant improvement of TVR for two reasons:

- Although $\hat{C}_{t_i}(\omega)$ does not approximate accurately, the zone $\{\omega | F_{t_i}(\omega) \geq C_{t_i}(\omega)\}$ is not differed by much, for otherwise we have our $\hat{C}_{t_i}$ very far from C_{t_i}. In the spirit of American option, we do not care how much the continuation value is worth, as long as it does not exceed F_{t_i}, as in such case we do not exercise.

- As a consequence we do not need update $\hat{\tau}_i$ that often as we did to t. By definition of this stopping time we have $\hat{\tau}_i = \hat{\tau}_{i+1}$ when $\hat{C}_i > F_{t_i}$.

Remark 2.5 (Choice of method??)**.** We need to estimate the conditional expectation for each C_{t_i}. As mentioned, it do not need to be very precise, but still we want some expedient way to compute it. The recipe on book used parametric method: namely to choose Black-Scholes model for the basket performance, and then do linear regression on it. I really have doubts for the validity of this method. The way to estimate C by linear regression is listed here. Therefore, I suggest use **Nonparametric regression**, namely we compute:

$$\mathbb{E}^{\mathbb{Q}}[Y_{i+1} \mid X_i = x] \approx \frac{\mathbb{E}^{\mathbb{Q}}[Y_{i+1}\delta_N(X_i - x)]}{\mathbb{E}^{\mathbb{Q}}[\delta_N(X_i - x)]} \tag{16}$$

with δ_N the Dirac sequence/smoothing kernel approximating a Dirac mass at zero: $\delta_h(x) = \frac{1}{h}e^{-(x/h)^2}$. Nonetheless, we need to tune the bandwidth h, as the sample path can perturb our value if the bandwidth is set too small.

A question here is when doing simulation, they seem only need to condition on $X_i = x$, while our filtration is $\mathcal{F}_t$ which should consist of all previous X_s information for $s \leq t$. I suspect this is due to the sin of approximation: we need to widen our bandwidth if more points is taken into account. Maybe you have some insights on this.

Remark 2.6 (bias-reduction)**.** A subtle point in estimating C_{t_i} is it used asset path data of the future $t \geq t_i$, so the result τ is not $\mathcal{F}_{t_i}$-stopping time. For this reason, we need to run LS for p_1 paths to get a strategy $\hat{\tau}$ and then draw p_2 paths that are stopped according to $\hat{\tau}$, namely we use a whole set of paths to measure $\hat{V}_0$ after τ is determined. Note $p_1 << p_2$.

Next we estimate the upper bound using **Andersen-Broadie algorithm(AB)**. The rough sketch is we first run a LS-algorithm to yield a sequence $\hat{\tau}_i$ of optimal stopping times. then estimate optimal strategy M^* by replacing V_{t_i} by $\hat{V}_{t_i}^{\hat{\tau}_i}$. Eventually plug our estimate $\hat{M}$ into RHS of formula in 2.1 to estimate $\hat{V}_0$.

As a byproduct of proving 2.1, we have the optimal strategy satisfies:

$$\begin{aligned} M^*_{t_i} - M^*_{t_{i-1}} &= D_{0t_i} V^{\tau_i}_{t_i} - \mathbb{E}^{\mathbb{Q}}[D_{0t_i} V^{\tau_i}_{t_i} \mid \mathcal{F}_{t_{i-1}}] \\ &= D_{0t_i} V^{\tau_i}_{t_i} - D_{0t_{i-1}} V^{\tau_{i-1}}_{t_{i-1}} - \mathbf{1}_{\tau_{i-1}=t_{i=1}} (\mathbb{E}^{\mathbb{Q}}[D_{0t_i} V^{\tau_i}_{t_i} \mid \mathcal{F}_{t_{i-1}}] - D_{0t_{i-1}} V^{\tau_{i-1}}_{t_{i-1}}) \end{aligned} \tag{17}$$

Note in this case the conditional expectation only need to be computed in the case $\hat{\tau}_{i-1}$ indicates immediate exercise. So the overall algorithm flows like such:

1. Simulate p_1 paths from 0 to T to get the strategy $\hat{\tau}_i$;
2. Simulate p_2 paths based on τ_i and LS-algorithm to compute lower bound;
3. Simulate p_3 paths from 0 to T. For each of these paths, and for each exercise date t_i:
   - if $\hat{\tau}_i \geq t_i$, then simulate p_4 sub-paths starting from t_i and estimate $D_{0t_i} \hat{V}^{\tau_i}_{t_i}$ by $\frac{1}{p_4} \sum_{j=1}^{p_4} D^j_{0\tau_i} F^j_{\tau_i}$
   - if $\hat{\tau} + i = t_i$, then estimate $D_{0t_i} \hat{V}^{\tau_i}_{t_i}$ by $D_{0t_i} F_{t_i}$. If $t_i < T$, then simulate p_5 independent subpaths starting from t_i and estimate $\mathbb{E}^{\mathbb{Q}}$-term by(WHY?):

$$\frac{1}{p_5} \sum_{j=1}^{p_5} D^j_{0\tau_{i+1}} F^j_{\tau_{i+1}} \tag{18}$$

4. Plugging all these in equation above, then averaging over p_3-paths after taking supremum.

Remark 2.7. The paths simulated altogether is of order $p_1 + p_2 + p_3(N-1)\max(p_4, p_5)$. Luckily for p_4, p_5, p_6 of order 10^2 the result provides accurate results.

3 improvements

There are several things we may work on:

- Add new underlying. This will create correlation problem. We can try to implement finite-difference, if have time.
- Try out some error/variance reduction. Example: stratified sampling method to reduce variance; Romberg extrapolation to reduce discretization error.
- parallelization: as we have large amounts of subpaths to generate, so we might as well want to devise a way to implement GPU resources. Colab can be a useful resource.