Устранение экспоненциальной сложности оценки стоимости бермудского опциона

Анастасия Миллер

СПбГУ, $6^{\text{ой}}$ семестр, 322 гр.
26 мая 2014 г.

1 Вступление

В книге [4] был предложен метод оценки американских опционов с конечным множеством дат погашения. Две оценки – смещённая вверх и смещённая вниз – получаются с помощью смоделированного дерева, которое ветвится при каждой возможности раннего погашения опциона. Оценки являются состоятельными (т.е. сходятся по вероятности к истинной цене опциона) и асимптотически несмещёнными.

Один из основных недостатков алгоритма – его экспоненциальная сложность. Здесь же предлагается несколько подходов, которые заменят экспоненциальную сложность полиномиальной с одновременным увеличением «случайности» алгоритма.

2 Общая идея алгоритма

Начиная с некоторого момента t_k, когда общее число состояний достигнет некоторого n, мы перестанем генерировать дочерние вершины ко всем состояниям. В следующий момент времени, t_{k+1}, мы будем иметь всё так же n состояний, а не bn. Этого можно достичь, если генерировать дочерние состояния не ко всем вершинам, а только к некоторым. К каким?

2.1 Анализ распределения состояний с помощью гистограммы

В том случае, когда состояние актива S является числом в $\mathbb{R}^1$, в качестве параметра X, распределение которого нас интересует, можно использовать само S, иначе можно использовать $h(S)$.

Деля интервал $\left[\min_{i\in 1:n} X_i; \max_{i\in 1:n} X_i + \frac{1}{n}\right)$ на k равных частей $[a_{k-1}, a_k)$, где $a_0 = \min_{i\in 1:n} X_i$, $a_k = \max_{i\in 1:n} X_i$, мы можем определить частоты $f_k = \#\left\{X_i | X_i \in [a_{k-1}, a_k)\right\}/n$ попадания событий в различные части отрезка. Из состояний, сгруппированных на отрезке $[a_{k-1}, a_k)$, мы также можем создать некоторый «средний арифметический» вектор, координаты которого будут являться средним арифметическим координат всех состояний, оказавшихся на данном отрезке, и уже для этого нового среднего состояния – представителя отрезка – генерировать дочерние вершины в количестве nf_k. Для всех состояний, оказавшихся в этом отрезке, дочерними вершинами будут являться все вершины, полученные от их представителя.

Таким образом, количество рассматриваемых состояний не увеличится. С другой стороны, этот метод предполагает хранение в памяти всего дерева, а не только непосредственно обсчитываемой ветки, как это предполагалось в исходной работе [3].

2.2 Кластеризация состояний

Для выделения родителей будущего поколения событий можно использовать не гистограммный подход, а кластеризацию существующего поколения. Так как состояния являются векторами в $\mathbb{R}^d$, в качестве метрики можно взять, например, «улучшенную» евклидову метрику в $\mathbb{R}^d$:

$$\mu\left(S_i, S_j\right) = \sqrt{\sum_{k=1}^{d} \frac{\left(s_k^i - s_k^j\right)}{c_k}},$$

где $c_k = \left|\max_{i\in 1:n} s_k^i - \min_{i\in 1:n} s_k^i\right|$, т.е. масштабирующий множитель, уравнивающий влияние различных компонент состояний на итоговое расстояние между ними.

Кластеризация может быть проведена по любому из известных алгоритмов, разберём один из наиболее простых и популярных алгоритмов кластеризации применительно к нашей задаче – алгоритм k-средних.

```
// назначаем центрами кластеров случайно выбранные S_k
for j ∈ 1 : k do
    Get(α)
    C_j = S_⌈nα⌉, ∀i ∈ 1 : j − 1C_i ≠ C_j
end
for j ∈ 1 : n do
    // центроидом для каждого состояния полагаем тот из
       центроидов, который ближе всего к данному состоянию
    S_j.centroid = argmin(i ∈ 1 : k, μ(S_j, C_i))
end
changed = true
repeat
    for j ∈ 1 : k do
        centroid = {i ∈ 1 : n | S_i.centroid = C_j}
        C_j = 1/#(centroid) · Σ_{i∈centroid} S_i
    end
    changed = false
    for j ∈ 1 : n do
        // пересчитываем принадлежность состояний центроидам
        oldcentroid = S_j.centroid
        S_j.centroid = argmin(i ∈ 1 : k, μ(S_j, C_i))
        changed = (oldcentroid == S_j.centroid)
    end
until changed = true
```

В этом случае набор центроидов $\{C_j\}_{j=1}^{k}$, можно использовать в качестве родителей для следующего поколения состояний.

Известные недостатки алгоритма k-средних, такие как сложность $O(2^n)$ в худшем случае, зависимость результатов от начального выбора центроидов (которая, впрочем, может быть частично устранена при модификации алгоритма, например, в версии k-means++, описанной в [1])

3 Выбор k

Выбор количества родителей для будущего поколения – отдельный вопрос. Скорее всего, количество родителей будет непосредственно влиять на величину доверительного интервала результата.

Список литературы

[1] David Arthur и Sergei Vassilvitskii. “k-means++: The Advantages of Careful Seeding”. В: *SODA*. 2007. URL: `http://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf`.

[2] Mark Broadie, Paul Glasserman и Gautam Jain. “Enhanced Monte Carlo estimates for american option prices”. В: *Journal of Derivatives* 5.1 (Fall) (1997), с. 25–44.

[3] M. Broadie и P. Glasserman. “Pricing American-style securities by simulation”. Английский. В: *Journal of Economic Dynamics and Control* 21 (1997), с. 1323–1352.

[4] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Английский. Springer, 2004.