Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Федеральное государственное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет»
Математико-механический факультет
Кафедра статистического моделирования

Отчёт о научно-исследовательской практике

Миллер Анастасия Александровна

Устранение экспоненциальной сложности оценки стоимости бермудского опциона

Научный руководитель:
д. ф.-м. н., профессор С. М. Ермаков

Санкт-Петербург
2014

Оглавление

Введение

Метод оценки американских опционов с конечным числом дат погашения, основанный на моделировании дерева событий (метод случайных деревьев), был предложен в ещё в 2004 году. Этот метод моделирует изменение состояния базового актива через случайные деревья, разветвляющиеся в каждой из возможных дат раннего погашения опциона. При анализе деревьев могут быть получены две оценки: смещённая вверх и смещённая вниз, являющиеся асимптотически несмещёнными и дающие доверительный интервал для истинной цены опциона.

Один из основных недостатков алгоритма — его экспоненциальная сложность. Рассмотрим алгоритм и реализацию одного из подходов, снижающего вычислительную сложность до полиномиальной с одновременным увеличением «случайности» алгоритма.

Глава 1

Метод случайных деревьев

1.1. Обозначения и умолчания

Будем строить модель на примере американского опциона с конечным числом дат погашения $t_1, \dots t_m$ (также называемого бермудским опционом). Мы также сузим класс решаемых нами задач до тех, в которых вся необходимая информация об активе, на который выписан рассматриваемый опцион, может быть представлена в виде Марковского процесса $X(t), t \in \{t_i\}_{i=1}^m$ со значениями в $\mathbb{R}^d$. Для уменьшения объёма текста будем обозначать $X(t_i) \equiv X_i$. Положим также $h_i(x)$ — размер выплаты по опциону в момент t_i при том, что $x = X_i$ и опцион не был исполнен до этого, $V_i(x)$ — стоимость опциона в момент t_i при том, что $x = X_i$.

Нетрудно видеть, что

$$V_m(x) = h_m(x), \tag{1.1}$$

$$V_{i-1}(x) = \max\{h_{i-1}(x), \mathsf{E}[V_i(X_i)|X_{i-1} = x]\} \tag{1.2}$$

— на каждом шаге мы выбираем наиболее выгодное решение. Здесь нас интересует значение $V_0(X_0)$.

1.2. Построение дерева и оценок

Вместо того, чтобы строить оценку, каким-либо образом стремящуюся к требуемому нами значению, мы построим две оценочные функции, оценивающие V_n сверху и снизу. Пусть $\hat{V}_n(b)$ и $\hat{v}_n(b)$ — такие оценки, зависящие от некоторого параметра b.

Метод случайного дерева основан на моделировании цепи $X_0, X_1, \dots X_n$ состояний актива. Зафиксируем параметр ветвления b. Из исходного состояния X_0 смоделируем b независимых следующих состояний $X_1^1, X_1^2, \dots X_1^b$, все с условием X_1. Для каждого X_1^i снова смоделируем b независимых последующих состояний $X_2^{i1}, \dots X_2^{ib}$. На m-ом шаге будем иметь b^m состояний, и это и есть источник основного недостатка этого метода —

его экспоненциальной алгоритмической сложности. Пример дерева можно увидеть на рис. 3.1.

1.2.1. Оценка сверху

Определим $\hat{V}_i^{j_1,j_2\cdots j_i}$, вдохновляясь (1.1). В последних вершинах (листьях) дерева зададим

$$\hat{V}_m^{j_1\cdots j_m} = h_m\left(X_m^{j_1\cdots j_m}\right). \tag{1.3}$$

Идя вверх по дереву, зададим

$$\hat{V}_i^{j_1\cdots j_i} = \max\left\{h_i\left(X_i^{j_1\cdots j_i}\right), \frac{1}{b}\sum_{j=1}^{b}\hat{V}_{i+1}^{j_1\cdots j_i j}\right\}. \tag{1.4}$$

С помощью индукции можно доказать, что наша оценка уклоняется вверх в каждом узле.

Теорема 1. $\forall i \in 1:n$

$$\mathsf{E}\left[\hat{V}_i^{j_1\cdots j_i}|X_i^{j_1\cdots j_i}\right] \geqslant V_i\left(X_i^{j_1\cdots j_i}\right).$$

Доказательство. В листьях дерева неравенство выполняется как равенство по определению.

Докажем, что если утверждение теоремы выполняется на $i+1$ шаге, то оно выполняется и на i. По определению

$$\mathsf{E}\left[\hat{V}_i^{j_1\cdots j_i}|X_i^{j_1\cdots j_i}\right] = \mathsf{E}\left[\max\left\{h_i\left(X_i^{j_1\cdots j_i}\right), \frac{1}{b}\sum_{j=1}^{b}\hat{V}_{i+1}^{j_1\cdots j_i j}\right\}|X_i^{j_1\cdots j_i}\right],$$

с помощью неравенства Йенсена ($\varphi\left(\mathsf{E}\left[X\right]\right) \leqslant \mathsf{E}\left[\varphi(X)\right]$) это можно оценить снизу:

$$\mathsf{E}\left[\max\left\{h_i\left(X_i^{j_1\cdots j_i}\right), \frac{1}{b}\sum_{j=1}^{b}\hat{V}_{i+1}^{j_1\cdots j_i j}\right\}|X_i^{j_1\cdots j_i}\right] \geqslant \max\left\{h_i\left(X_i^{j_1\cdots j_i}\right), \mathsf{E}\left[\frac{1}{b}\sum_{j=1}^{b}\hat{V}_{i+1}^{j_1\cdots j_i j}|X_i^{j_1\cdots j_i}\right]\right\};$$

в силу того, что $\forall\, j \in 1:b \quad X_{i+1}^{j_1\cdots j_i j}$ — независимые одинаково распределённые случайные величины (и их математическое ожидание одинаково), $\mathsf{E}\left[\frac{1}{b}\sum_{j=1}^{b}\hat{V}_{i+1}^{j_1\cdots j_i j}\right] = \frac{1}{b}\sum_{j=1}^{b}\mathsf{E}\hat{V}_{i+1}^{j_1\cdots j_i j} = \mathsf{E}\hat{V}_{i+1}^{j_1\cdots j_i 1}$, а в силу индукционного предположения

$$\max\left\{h_i\left(X_i^{j_1\cdots j_i}\right), \mathsf{E}\left[\hat{V}_{i+1}^{j_1\cdots j_i 1}|X_i^{j_1\cdots j_i}\right]\right\} \geqslant \max\left\{h_i\left(X_i^{j_1\cdots j_i}\right), V_i\left(X_i^{j_1\cdots j_i}\right)\right\}.$$

Таким образом, $\mathsf{E}\left[\hat{V}_i^{j_1\cdots j_i}|X_i^{j_1\cdots j_i}\right] \geqslant \max\left\{h_i\left(X_i^{j_1\cdots j_i}\right), V_i\left(X_i^{j_1\cdots j_i}\right)\right\}$. □

Мы также доказываем, что $\hat{V}_i^{j_1\ldots j_i}$ сходится по вероятности к $V_i\left(X_i^{j_1\ldots j_i}\right)$ при $b\to\infty$. В листьях дерева это очевидно ($\hat{V}_m^{j_1\ldots j_m} = h_m\left(X_m^{j1\ldots j_m}\right)$ по определению), на $i-1$ шаге цена удержания опциона $\frac{1}{b}\sum_{j=1}^{b}\hat{V}_{i+1}^{j_1\ldots j_i j}$ является средним арифметическим независимых одинаково распределённых случайных величин и сходится по закону больших чисел. Сходимость распространяется и на саму оценку в силу непрерывности операции взятия максимума. Используя тот факт, что $\forall\, a, c_1, c_2 \in \mathbb{R}\ |\max(a,c_1) - \max(a,c_2)\,| \leqslant |c_1 - c_2|$, мы получаем

$$\left|\hat{V}_i^{j_1\cdots j_i} - V_i\left(X_i^{j_1\cdots j_i}\right)\right| \leqslant \frac{1}{b}\sum_{j=1}^{b}\left|\hat{V}_{i+1}^{j_1\cdots j_i j} - \mathsf{E}\left[V_{i+1}\left(X_{i+1}^{j_1\cdots j_i j}\right)|X_{i+1}^{j_1\cdots j_i}\right]\right|,$$

что позволяет нам вывести из сходимости на $i+1$ шаге сходимость на i шаге. Подробное описание оценок, их асимптотик и доверительных интервалов можно найти в [1].

Более того, оценка $\hat{V}_0$ является асимптотически несмещённой, т.е. $\mathsf{E}\hat{V}_0 \to V_0(X_0)$.

1.2.2. Оценка снизу

Значения оценки сверху в каждый момент времени — это выбор максимума из стоимости опциона при его немедленном исполнении и математического ожидания стоимости удержания опциона. Но стоимость удержания опциона рассчитывается, исходя из дочерних узлов дерева состояний актива,то есть оценка сверху рассчитывается, опираясь на информацию о будущем. Чтобы убрать ошибку, связанную с этим, нам необходимо отделить механизм принятия решения о исполнении/удержании опциона от значений, полученных после принятия решения об удержании опциона.

По сути, нам нужно оценить $\max\{a, \mathsf{E}Y\}$ с помощью b независимых одинаково распределённых реализаций случайной величины Y для некоторой константы a и случайной величины Y. Оценка $\max\left\{a, \bar{Y}\right\}$ (где $\bar{Y}$ — среднее значение выборки) является оценкой сверху, так как $\mathsf{E}\max\left\{a, \bar{Y}\right\} \geqslant \max\left\{a, \mathsf{E}\bar{Y}\right\} = \max\{a, \mathsf{E}Y\}$, что мы и использовали в построеии нашей оценки сверху.

Разделим наше множество реализаций $\{Y_i\}_{i=1}^{b}$ случайной величины Y на два независимых подмножества и вычислим их средние значения $\bar{Y}_1$ и $\bar{Y}_2$. Если положить

$$\hat{v} = \begin{cases} a, & \text{если } \bar{Y}_1 \leqslant a \\ \bar{Y}_2, & \text{иначе} \end{cases}, \tag{1.5}$$

мы отделим процесс принятия решения об исполнении или удержании опциона от оценки его стоимости (за решение будет отвечать $\bar{Y}_1$, за оценку — $\bar{Y}_2$). При этом оценка $\hat{v}$ является оценкой снизу:

$$\mathsf{E}\hat{v} = \mathsf{P}\left(\bar{Y}_1 \leqslant a\right) a + \left(1 - -- \mathsf{P}\left(\bar{Y}_1 \leqslant a\right)\right) \mathsf{E}Y \leqslant \max\left\{a, \mathsf{E}Y\right\}. \tag{1.6}$$

Нижняя оценка по [2] Пусть «отвечающим за принятие решения» подмножеством будут все реализации, кроме одной, а «оценивающее» множество будет состоять из одной оставшейся реализации. Возьмём математическое ожидание этой величины, т.е. положим в листьях дерева значение оценки

$$\hat{v}_m^{j_1 j_2 \cdots j_m} = h\left(X_m^{j_1 j_2 \cdots j_m}\right), \tag{1.7}$$

для промежуточных узлов определим

$$\hat{v}_{ik}^{j_1 j_2 \cdots j_i} = \begin{cases} h\left(X_i^{j_1 j_2 \cdots j_i}\right), & \text{если } \frac{1}{b-1} \sum_{j=1, j\neq k}^{b} \hat{v}_{i+1}^{j_1 j_2 \cdots j_i j} \leq h\left(X_i^{j_1 j_2 \cdots j_i}\right) \\ \hat{v}_{i+1}^{j_1 j_2 \cdots j_i k}, & \text{иначе} \end{cases}, \tag{1.8}$$

и оценку положим равной

$$\hat{v}_i^{j_1 j_2 \cdots j_i} = \frac{1}{b} \sum_{k=1}^{b} \hat{v}_{ik}^{j_1 j_2 \cdots j_i}.$$

Таким образом, мы имеем дерево с $\sum_{k=1}^{m} b^k = \frac{b(b^m - 1)}{b-1} = O\left(b^m\right)$ вершинами и две оценки искомой величины по этому дереву.

Глава 2

Устранение экспоненциальной сложности

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

$$f_k = \frac{1}{n}\#\left\{X_i | X_i \in [a_{k-1}, a_k)\right\}$$

попадания событий в различные части отрезка. Из состояний, сгруппированных на отрезке $[a_{k-1}, a_k)$, мы также можем создать некоторый «средний арифметический» вектор, кооринаты 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 $n \cdot f_k$. 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 [1].

Гистограммный анализ «среза» распределения проходит за $O(n)$ времени и за $O(n)$ по памяти, что приводит нас к общему времени работы $O(mn)$ и общей затраченной памяти $O(mn)$, где m — число дат погашения опциона, включая первую и последнюю.

Глава 3

Реализация

В качестве параметра, распределение которого будет анализироваться, я использовала цену актива (реализацию винеровского случайного процесса, где каждое следующее состояние получается из предыдущего как $p_0 \cdot (1 + a \cdot \triangle t + \sigma \cdot \varepsilon \cdot \sqrt{\triangle t})$, где p_0 — цена актива в предыдущий момент времени, $\triangle t = 1/s$, a и σ означают доходность и волатильность цены акции соответственно и являются константами, ε — случайная величина со стандартным нормальным распределением).

На рис. 3.1 можно видеть, как выглядит генерируемое исходным методом дерево.

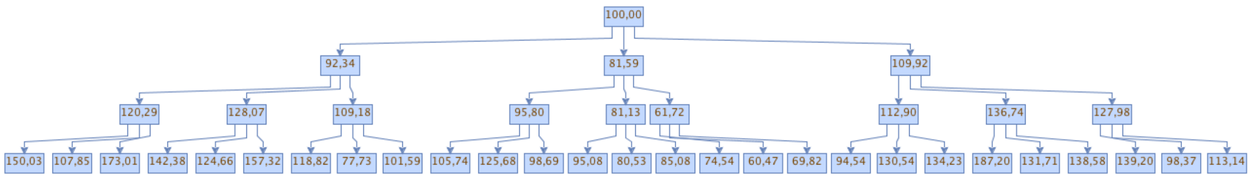

Рис. 3.1. Дерево, генерируемое при использовании метода, описываемого Глассерманом (цифры в узлах — стоимость актива)

В своей реализации я разделяю генерацию дерева и подсчёт оценок, ему соответствующих, так как оценки, в отличие от исходных деревьев, у меня не отличаются от оценок у Броади и Глассермана. Вначале существовала надежда сравнивать «полные» и «урезанные» деревья, но она не оправдалась из-за слишком больших требований к памяти у классических деревьев.

Момент, после которого стоит переходить на линейную модель генерации дочерних вершин, определился как $k = \left\lfloor {}^{\log n}/_{\log b} \right\rfloor$ (b — количество ветвей у узла в «экспоненциальном режиме», n — ширина дерева в «линейном режиме»). Реализацию можно увидеть в приложении A.1.

Дерево, генерируемое усечённым образом, можно увидеть на рис. 3.2.

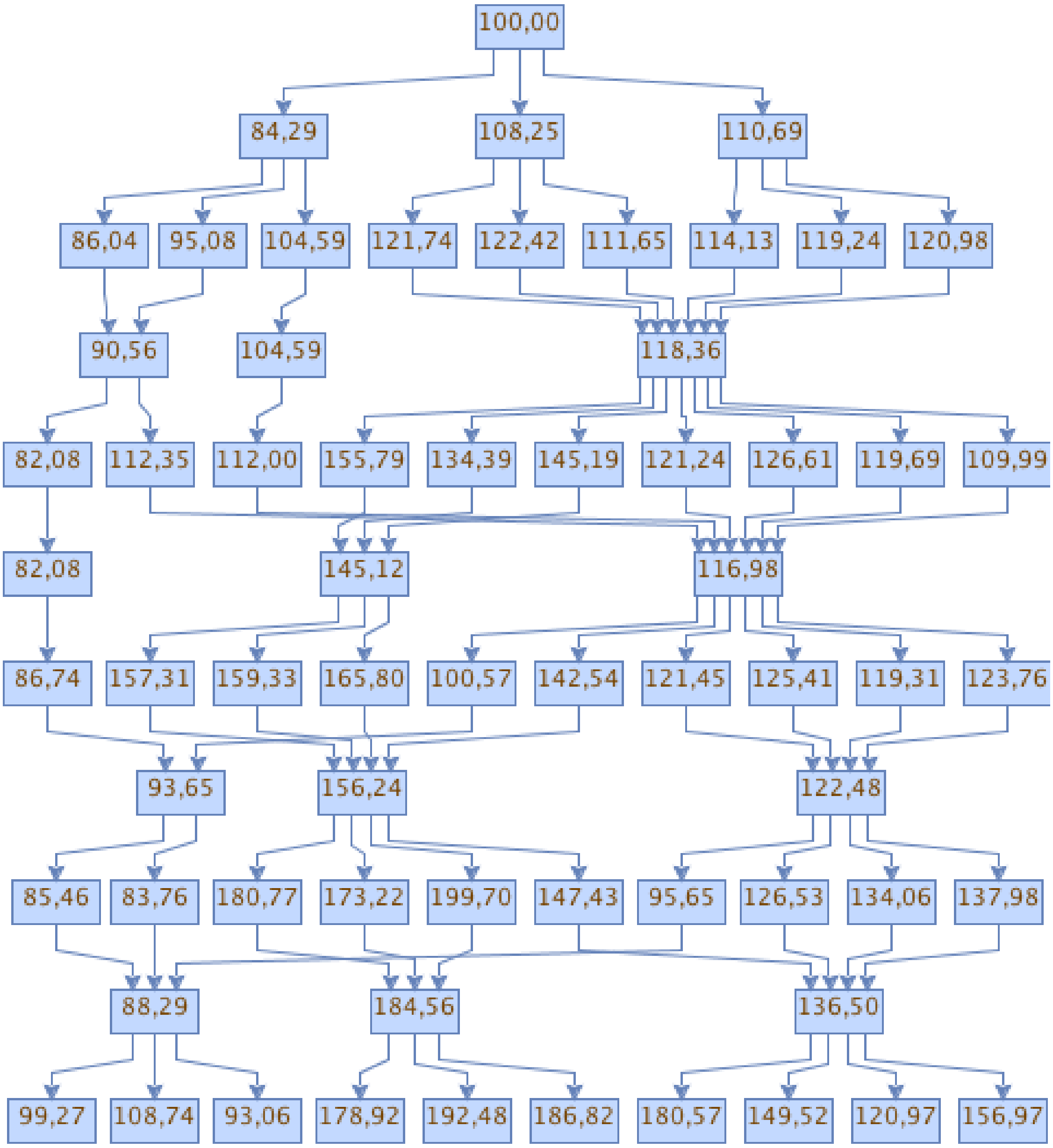

Рис. 3.2. Дерево, генерируемое усечённым методом (цифры — стоимость актива; ширина дерева $n = 10$, количество секторов $k = 3$)

Глава 4

Результаты

Целью увеличения доступного для обсчёта числа дат исполнения было максимально приблизиться к американскому опциону (опциону с возможностью исполнения в любой момент в оговорённом промежутке времени). Неизвестными факторами (поведение которых не было очевидным на стадии создания упрощённого метода) были ширина дерева n и количество «столбцов гистограммы» k. Также было неясно, существует ли сходимость метода, сопоставимая со сходимостью исходного метода.

Испытания сходимости метода были проведены по алгоритму 1.

```
startSteps = 50
for i ∈ 1 : 100 do
    x_1 = ∞
    x_0 = −∞
    step = 0
    while |x_1 − x_0| > ϵ and step<1000 do
        x_0 = x_1
        x_1 = estimateAsset(steps=startSteps+step, width=50, sectors=k)
        step += 1
    end
    if step == 1000 then
        в этом испытании алгоритм не сошёлся
    else
        в этом испытании алгоритм сошёлся
    end
end
```

Алгоритм 1: Проверка сходимости оценки в испытании

Для верхней оценки стоимости опциона результаты можно увидеть на рис. 4.1, для нижней оценки — на рис. 4.2.

Вероятность (выборочная) сходимости верхней и нижней оценки представлена на гра-

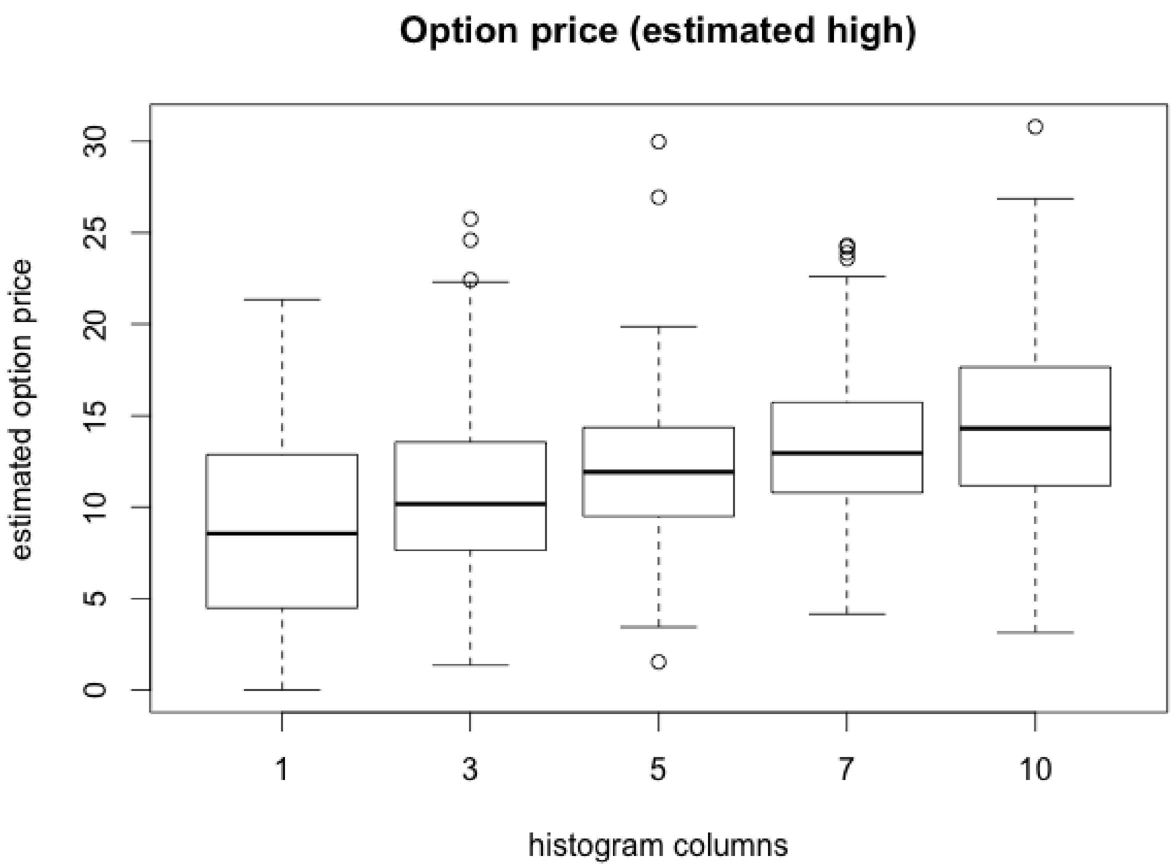

Рис. 4.1. Распределение верхней оценки стоимости опциона

фиках 4.3 и 4.4 соответственно.

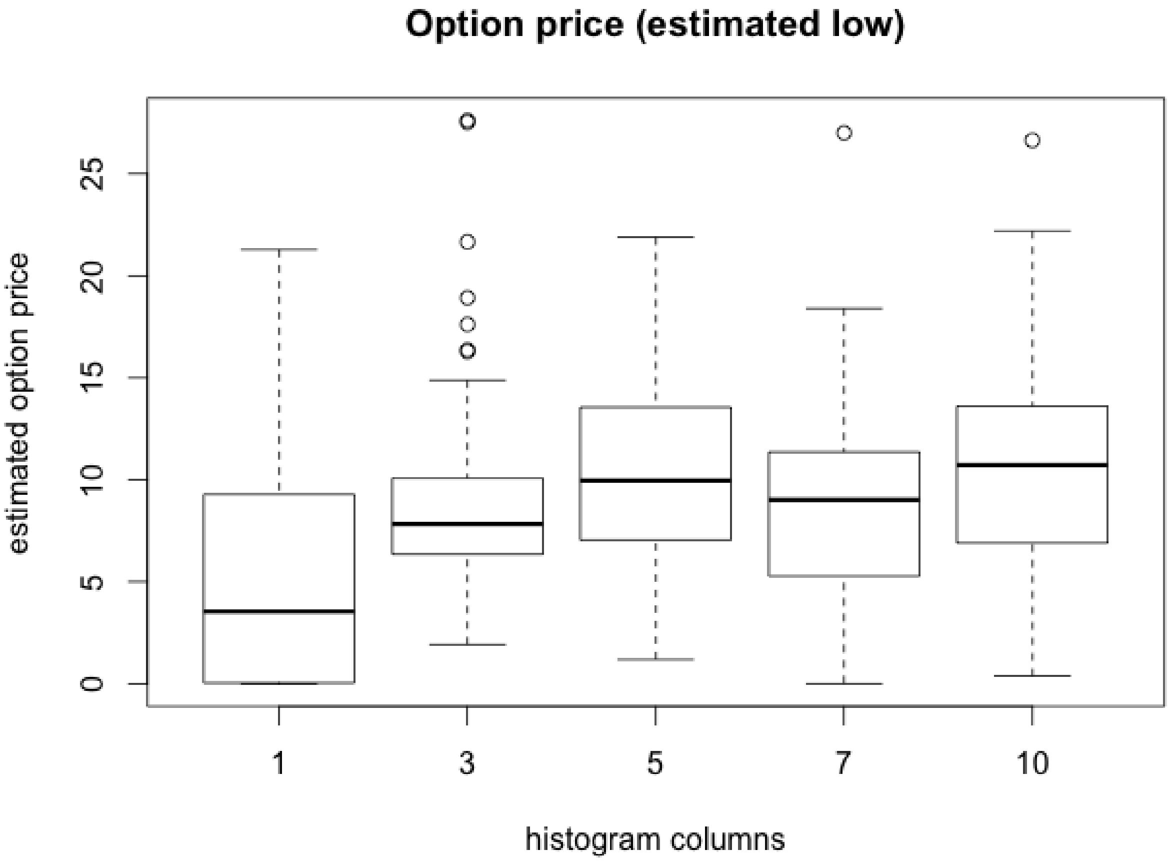

Рис. 4.2. Распределение нижней оценки стоимости опциона

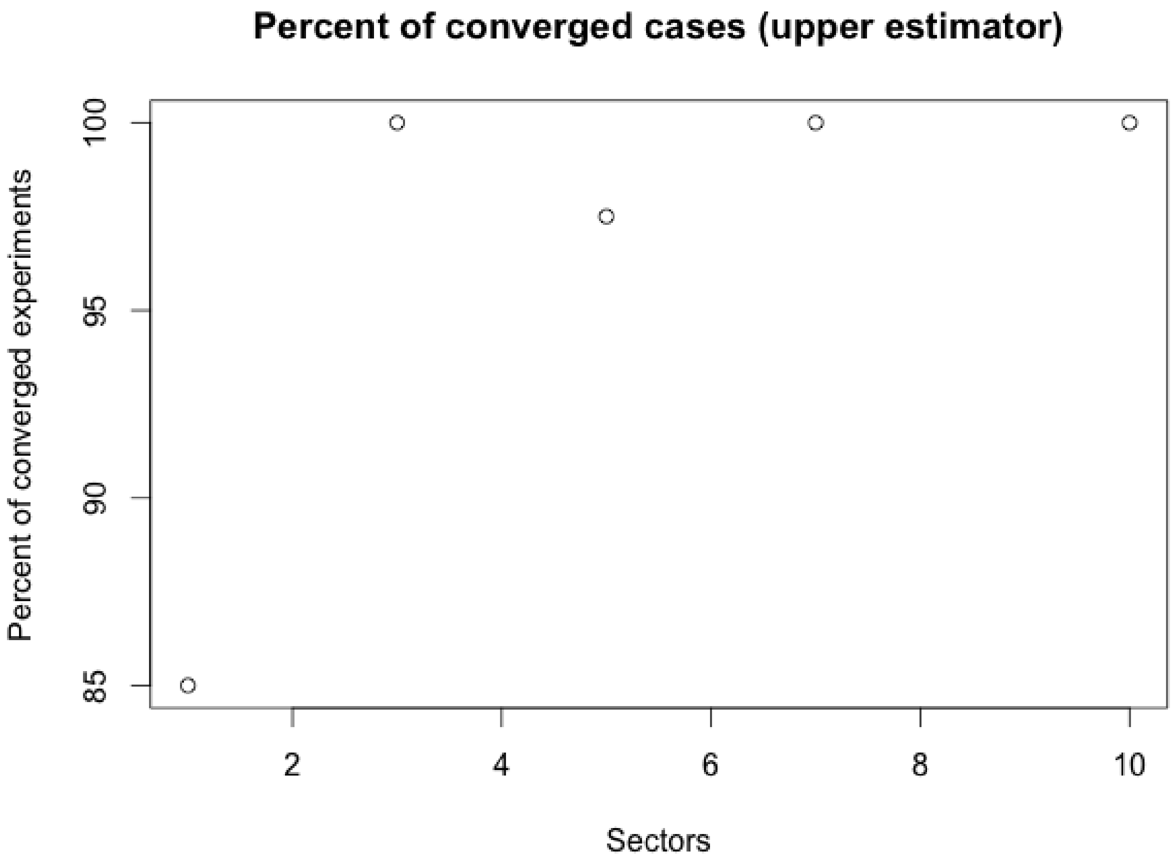

Рис. 4.3. Процент случаев, в которых верхняя оценка сошлась, по отношению к общему числу испытаний

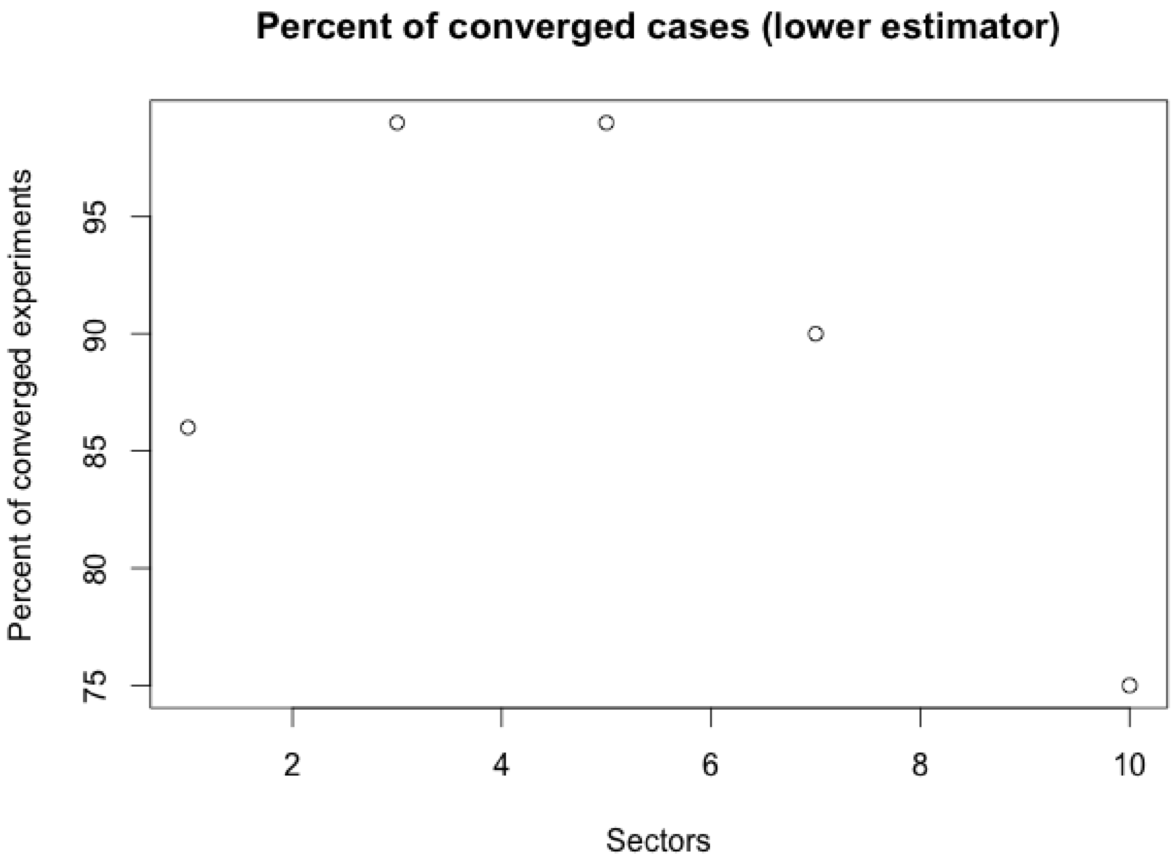

Рис. 4.4. Процент случаев, в которых нижняя оценка сошлась, по отношению к общему числу испытаний

Заключение

Как видно, оценки имеют большую дисперсию, причём если для нижней оценки эксперименты показывают, что оптимальное число подмножеств — 3, то для верхней оценки локальный минимум не очевиден.

Дальнейшие планы

1. Закончить рассмотрение оценки по гистограмме, в т.ч. найти аналитически математическое ожидание оценки (похожий случай уже рассмотрен в [3]).

2. Рассмотреть оценку по кластерам (предполагаемый алгоритм кластеризации рассмотрен в [4]).

3. Рассмотреть другие оценки.

Список литературы

1. Broadie M., Glasserman P. Pricing American-style securities by simulation // Journal of Economic Dynamics and Control. — 1997. — Vol. 21. — P. 1323–1352.
2. Glasserman Paul. Monte Carlo Methods in Financial Engineering. — Springer, 2004.
3. Ермаков Сергей Михайлович. Метод Монте-Карло и смежные вопросы. — Наука, 1975.
4. Arthur David, Vassilvitskii Sergei. k-means++: The Advantages of Careful Seeding // SODA. — 2007. — URL: http://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf.
5. Broadie Mark, Glasserman Paul, Jain Gautam. Enhanced Monte Carlo estimates for american option prices // Journal of Derivatives. — 1997. — Vol. 5, no. 1 (Fall). — P. 25–44.

Приложение А

Реализация на Java

Листинг А.1. Генерирование дерева состояний актива, на который выписан опцион

```
public static ImitatedAsset generateAssetByHistogram(int width, int branch,
    int steps, int sectors, double initialPrice){
    timedelta = 1. / steps;
    int expSteps = (int) Math.floor(Math.log(width) / Math.log(branch));
    ImitatedAsset[] nodes = new ImitatedAsset[width];
    ImitatedAsset ans = generateTreeAssetsToModeling(branch, expSteps,
        initialPrice, nodes);

    ImitatedAsset[] new_nodes = generateFirstRow(width, sectors, nodes);
    nodes = new_nodes;

    // +1 because of one step that was done outside the cycle
    for (int step = expSteps + 1; step < steps; step++) {
        new_nodes = generateRow(width, (step + 1 == steps), sectors, nodes);
        nodes = new_nodes;
    }
    return ans;
}
private static ImitatedAsset[] generateRow(int width, boolean lastRow, int
    sectors, ImitatedAsset[] nodes) {
    ImitatedAsset[] new_nodes;
    sortArrayWithNulls(nodes);
    double sector = getSectorWidth(sectors, nodes);
    double min = extremalValue(nodes, -1);
    double sum = 0;
    int amount = 0;
    int k = 0;
    new_nodes = new ImitatedAsset[width];
    for (int j = 0; j < width; j++){ // iterating over {{nodes}}
        if (nodes[j].price > min + (k+1) * sector) { // reached the end of
            the sector
            generateBlock(nodes, new_nodes, lastRow, j-amount, j, amount,
                sum/amount);
```

```
                k++;
                amount = 0;
                sum = 0;
            }
            sum += nodes[j].price;
            amount++;
        }
        generateBlock(nodes, new_nodes, lastRow, width-amount, width, amount,
            sum/amount);
        return new_nodes;
    }
    private static void generateBlock(ImitatedAsset[] nodes, ImitatedAsset[]
        new_nodes, boolean lastRow, int start, int end, int children, double
        price, int new_start){
        ImitatedAsset asset = new ImitatedAsset(price, children, false); //
            intermediate asset will definitely have children
        for (int i = start; i < end; i++ ){ // assign average node as a child to
            the previous generation
            nodes[i].children[0] = asset;
        }
        for (int i = 0; i < children; i++){ // generating new nodes
            asset.children[i] = new ImitatedAsset(getRandomPrice(asset.price), 1,
                lastRow);
            new_nodes[new_start+i] = asset.children[i];
        }
    }
```