Random Tree Method

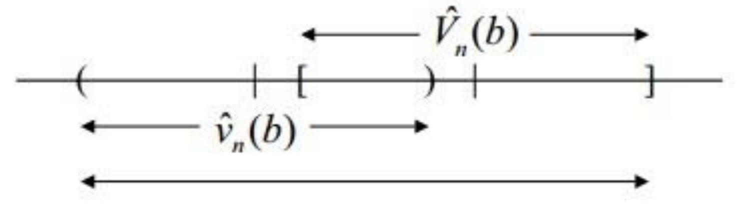
Monte Carlo Methods in Financial Engineering

What is it for?

- solve full optimal stopping problem & estimate value of the American option
- simulate paths of underlying Markov chain
- produces two consistent estimators: one **biased high** and one **biased low converging to the wanted Value V_0**
- renders a **confidence interval for the „true“ Value V_0**
- control error
- **but only for small m** (e.g. m = 5), limits the scope of the method
- computing time grows exponentially (in the number of exercise dates)

Find the interval

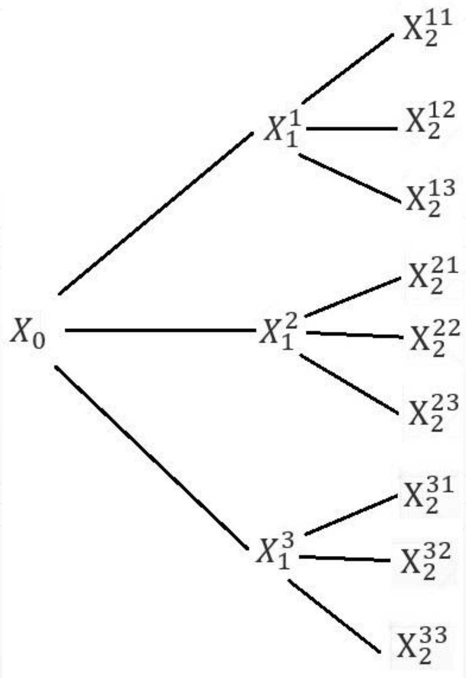
- simulation parameter b, time parameter m
- $\hat{V}(b)$ and $\hat{v}(b)$ sample means of n independent replications for some V_0
- so that: $E[\hat{V}(b)] \geq V_0 \geq E[\hat{v}(b)]$
- we need some halfwidths $L_n(b)$ for low bias and $H_n(b)$ for high bias
- confidence interval $\widehat{v_n}(b) - L_n(b), \widehat{V_n}(b) + H_n(b)$



- for $\mathrm{n} \rightarrow \infty$ the halfwidths shrink to 0 and our estimators come really close to V_0 for b → ∞

The Tree

- for simulating a tree of paths need underlying *Markov chain* $X_0, \ldots, X_m$
- *branching parameter b, time/date* $0, \ldots, m$
- from initial state X_0 simulate *b i.i.d. successor states* $X_1^1, \ldots, X_1^b$ all having law of X_1. From each X_1^i simulate *b i.i.d. successors* $X_2^{i1}, \ldots, X_2^{ib}$ with law of X_2 $given\, X_1 = X_1^i$.
  From each $X_2^{i_1 i_2}$ generate *b more successors* $X_3^{i_1 i_2 1}, \ldots, X_3^{i_1 i_2 b}$ and so on…
- denote generic node at step time i by $X_i^{j_1, \ldots, j_i}$
- superscript indicates, that the node is reached by following the j_1 branch out of X_0, the j_2 branch aut of $X_1^{j_1}$ etc.
- not essential, that branching parameter b is fixed across the time steps, but it‘s a convenient simplification



High Estimator

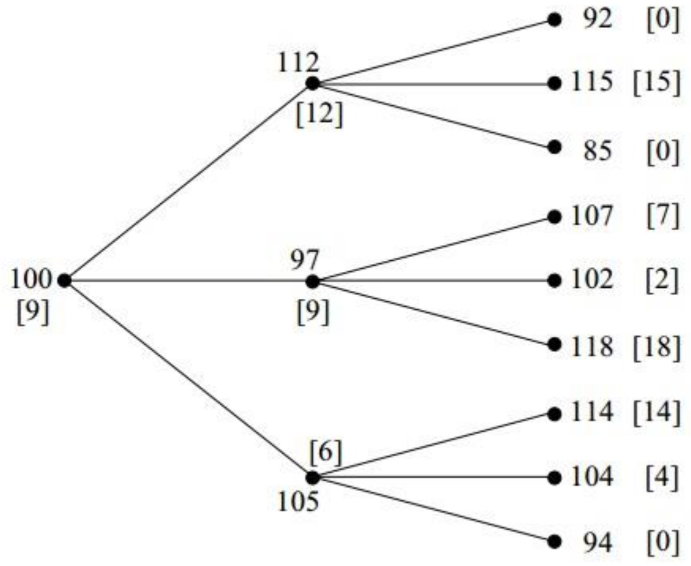
- from random tree define high & low estimators at each node by **backward induction**: $V_{i-1}(x) = \max\{\, h_{i-1}(x),\ E[V_i(X_i)|X_{i-1} = x]\,\}$

- $\mathbf{h_i(x) = (x - 100)^+}$ is the discounted **payoff function** with strike price 100 at the i'th exercise date and $\boldsymbol{V_m = h_m}$ is satisfied from the discounted **option value**

- **<u>high estimator:</u>** $\boldsymbol{V_i(x)} = \max\{\, h_i(x),\ E[V_{i+1}(X_{i+1})|X_i = x]\,\}$ $\quad i = 0, \dots, m-1$

- write for the high estimator: $\hat{V}_i^{\,j_1 \dots j_i}$ for each node $X_i^{j_1,\dots,j_i}$. Set at the **terminal nodes** $\hat{V}_m^{\,j_1 \dots j_m} = h_m(\,X_i^{j_1,\dots,j_m})$. Working backward: $\hat{V}_i^{\,j_1 \dots j_i} = \max\{\, h_i\left(\,X_i^{j_1,\dots,j_i}\right), \frac{1}{b}\sum_{j=1}^{b} \hat{V}_{i+1}^{\,j_1 \dots j_i j}\,\}$

- first term inside max.: immediate exercise
  second term: continuation value
  ➡ choosing the max the estimator is deciding whether to exercise or continue!



High Estimator – confidence interval

- under modest conditions **each $\hat{V}_i^{j_1..j_i}$ converges** in propability (and in norm) as b $\to \infty$ to the **true value** $V_i\left(X_i^{j_1,\ldots,j_i}\right)$ $given\ h_i\left(X_i^{j_1,\ldots,j_i}\right)$, this holds at all terminal nodes because V_m is initialized to $h_i = V_m$. (The continuation value at the $(m-1)$th step is the average of i.i.d. replications and converges by the law of lage numbers.)

- primaly interested in $\widehat{V_0}$, the estimate of the option price at the current time and state

- let $\bar{V}_0(n,b)$ denote the sample mean $(\bar{x})$ of the n replications of $\bar{V}_0$, let $s_V(n,b)$ denote their sample standard deviation. With $z_{\frac{\alpha}{2}}$ denoting the 1- $\frac{\alpha}{2}$ quantile of the normal distribution

➡ $\boldsymbol{\bar{V}_0(n,b) \pm z_{\frac{\alpha}{2}} \frac{s_V(n,b)}{\sqrt{n}}}$ ⬅ (first half of wanted interval for the true value V_0)

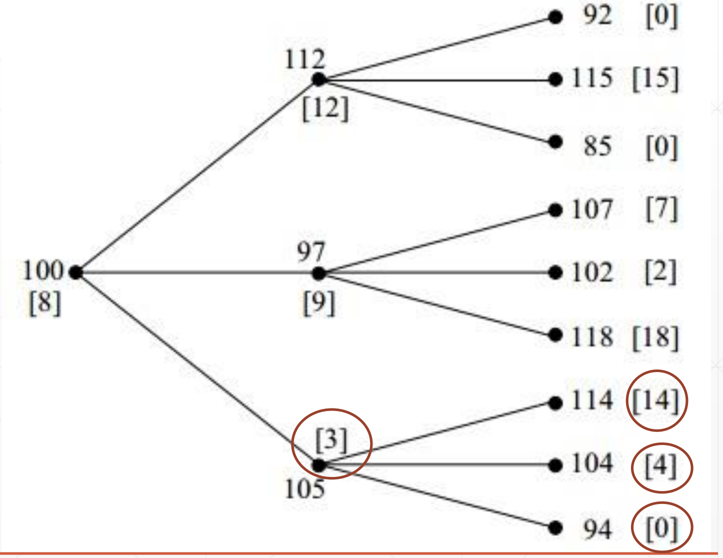
- provides (for large n) an asymptotically valid 1- $\frac{\alpha}{2}$ - confidence interval for $E[\widehat{V_0}]$

- the high estimator is based on successor nodes (unfairly peeking into future)

Low Estimator

- to remove „successor-based“ source of bias: seperate exercise decision from continuation value
- at all **terminal nodes** set: $\hat{v}_m{}^{j_1 \dots j_m} = h_m(X_i^{j_1,\dots,j_m})$. At nodes $j_1, \dots, j_i$ at time step i and for $k = 1, \dots, b$ set for the **low estimatior**: $$\hat{v}_{ik}^{j_1 \dots j_i} = \begin{cases} h_i(X_i^{j_1 \dots j_i}) & \text{if } \frac{1}{b-1} \sum_{j=1 j \neq k}^{b} \hat{v}_{i+1}^{j_1 \dots j_i j} \leq h_i(X_i^{j_1 \dots j_i}) \\ \hat{v}_{i+k}^{j_1 \dots j_i k} & \text{otherwise} \end{cases}$$
- exsample: leave out first successor node: $\frac{1}{2}(4+0) = 2 \leq 5 = h_i(X_i^{j_1,\dots,j_i})$, so we exercise and get 5
  leave out second one: $\frac{1}{2}(14+0) = 7 \geq 5$, continue and get 4
  leave out third: : $\frac{1}{2}(14+4) = 9 \geq 5$, continue and get 0

  averaging payoffs: $\frac{1}{3}(5+4+0) = 3$, so we get a low estimator of 3 at that node. ([3])



Low Estimator – confidence interval

- $\hat{\boldsymbol{v}}_{\mathbf{0}}$ **converges** in propability and in norm as $\mathrm{b} \rightarrow \infty$ to the **true value** $\boldsymbol{V}_{\mathbf{0}}(\boldsymbol{X}_{\mathbf{0}})$
- $\widehat{v_0}$ estimator of the option price at the current time and state
- let $\bar{v}_0(n,b)$ denote the sample mean $(\bar{x})$ of the n replications of $\bar{v}_0$, let $s_v(n,b)$ denote their sample standard deviation. With $z_{\frac{\alpha}{2}}$ denoting the 1-$\frac{\alpha}{2}$ quantile of the normal distribution

$$\Rightarrow \quad \boldsymbol{\bar{v}_0(n,b) \pm z_{\frac{\alpha}{2}} \frac{s_v(n,b)}{\sqrt{n}}} \quad \Leftarrow$$

(second half of wanted interval for the true value V_0)

- finally found interval for $\boldsymbol{V}_{\mathbf{0}}(\boldsymbol{X}_{\mathbf{0}})$:

$$\left(\bar{v}_0(n,b) \pm z_{\frac{\alpha}{2}} \frac{s_v(n,b)}{\sqrt[2]{n}}, \bar{V}_0(n,b) \pm z_{\frac{\alpha}{2}} \frac{s_V(n,b)}{\sqrt[2]{n}}\right)$$

by increasing n and b see that $E\left[\hat{V}_0\right]$ and $E[\hat{v}_0]$ approach $\boldsymbol{V}_{\mathbf{0}}(\boldsymbol{X}_{\mathbf{0}})$ and the confidence interval can be made as tight as we need.

- simple technique for error control

Implementation

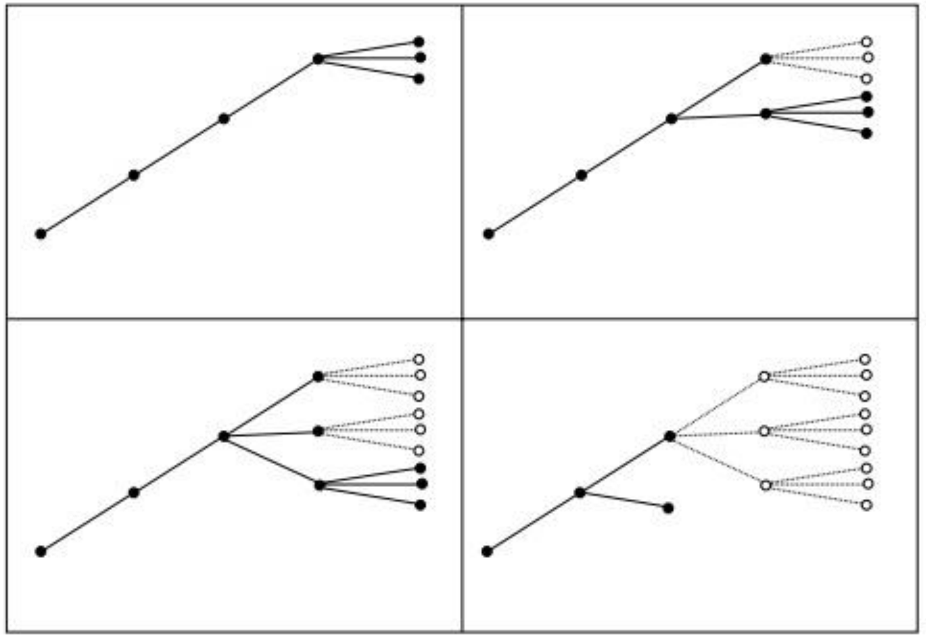
- wont generate naive all $\boldsymbol{m^b}$ nodes. Each node depends only on subtree
  ➡ reduce storage requirements of the method, so it‘s never necessary to store more than $\boldsymbol{mb+1}$ **nodes at a time**.

Depth-First Processing

- indices $j_1, \ldots, j_i$ taking values in $\{1, \ldots, b\}$. The string $j_1, \ldots, j_i$ labels the node reached by following the branches…

- in depth-first algorithm we follow a single branch rather than generating all branches at once

- like: genertae 1, 11, 111, 1111. At this point we reached the terminal step and can go no deeper, so generate nodes 1112,…,111b. From these values we calculate high and low biased estimators at (for) node 111. Next, generate 112 and it‘s successors: 1121,1122,…,112b using them to calculate high and low estimators at 112, then discard and repeat…
  113,..,11b ➞ node 11
  12,…,1b ➞ node 1
  2,…,b ➞ root node



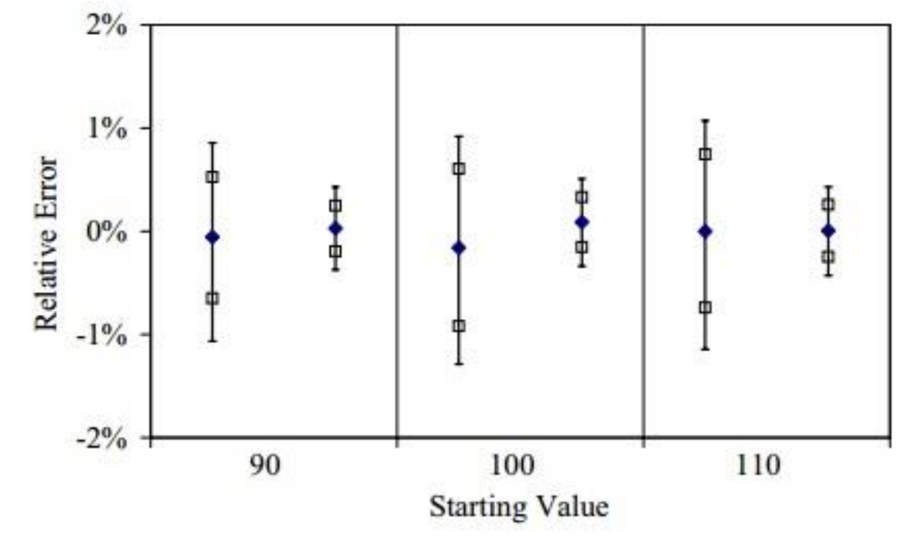
Prunning and Variance Conrol

- branching is needed only *when the optimal exercise decision is unknown*. If we know at which time it is optimal not to exercise, than it *would already suffice to generate just a single branch out of that node*. Working, then, backward through the tree the value we assign to that node is for both (high and low) estimators the (discounted) value of the corresponding estimator at the unique successor node.

- And **this** is an ordinary simulation to price an European option, where we *never have the choice to exercise early*. For that we <u>compare European options with American options </u>and use antithetic variates. If the value of the *European option exceeds the payoff of the American option it is optimal to continue*. In case, that the European option can be prised quickly, then the size of the tree *reduces by a factor of b.*

- e.g. pruning, antithetic branching and control variate interval closer to the exact value.
  - first: 100 replications; but second: several thousands
  - GBM=0.30 || 5=% || strike price=100 || b=50 || 90% coverage

- random tree method producing high and low estimators and render **estimates exact to within about 1%!**



Thank you

for your attention!

Jana Frank