Course Project: Implied Binomial Tree

Numerical Methods in Quantitative Finance

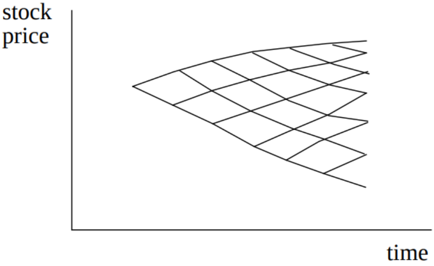
Li Hao
matv128@nus.edu.sg

Implied Binomial Tree

Project Objective

Given an implied volatility surface, construct and implement an implied binomial tree (IBT) and its pricer, so that we can price exotic products consistently with the implied volatility surface.

We will follow the Derman and Kani's algorithm [Derman and Kani, 1994] to construct the implied binomial tree.



Implied Binomial Tree Construction

- Implied volatility surface $\sigma_{iv}(t, K)$ are given
- To construct n-step tree, the time interval $[0, T]$ is equally spced: $\Delta t = \frac{T}{n}$
- Risk-free interate rate r
- Implied binomial tree is constructed by forward induction: from the constructed tree up to the $(k-1)$-th time step, we need to derive the k-th time step. The parameters to construct are
  - The position of the nodes $S_{k,i}$, for $i \in 0, 1, \ldots, k \quad \rightarrowtail k+1$ unknowns
  - The transition probability $p_{k,i}$: the probability that the stock price goes from $S_{k-1,i}$ to $S_{k,i} \quad \rightarrowtail k$ unknowns
  - Knowing $p_{k,i}$, the probability that the stock price goes from $S_{k-1,i}$ to $S_{k-1,i+1}$ is just $1 - p_{k,i}$
  - In total we have $2k+1$ unknowns for the k-th step induction

Implied Binomial Tree Construction

Now let's look at how many constraints we have

1. The forward price conditional on each node, should satisfy the risk-neutral condition:

$$F_{k-1,i} = \mathbb{E}[S_k | S_{k-1,i}] = S_{k-1,i} e^{r\Delta t} = p_{k,i} S_{k,i} + (1 - p_{k,i}) S_{k,i+1} \quad (1)$$

   In other words

$$p_{k,i} = \frac{S_{k-1,i} e^{(r-q)\Delta t} - S_{k,i+1}}{S_{k,i} - S_{k,i+1}} = \frac{F_{k-1,i} - S_{k,i+1}}{S_{k,i} - S_{k,i+1}} \quad (2)$$

   Here we have k constraints.

2. The option price struck at $S_{k-1,i}$ and expiry at k-th step (so time to maturity is $k\Delta t$), should match the option price given by the implied volatility surface. Here we also have k constraints. Let us elaborate on this.

Matching Option Prices

- Call option price $C(K, k\Delta t)$ and put option price $P(K, k\Delta t)$ can be obtained from querying the implied volatility surface, and plug into Black-Scholes formulae. These prices are our targets to match.
- The way to obtain option price from the implied binomial tree is through the Arrow-Debreu prices, denoted as $\lambda_{k,i}$, at each node.
- Arrow-Debreu price $\lambda_{k,i}$ is the discounted risk-neutral probabilities of each node. In the binomial tree model, it is also the price of an option that pays 1 unit payoff at time k, if and only if the stock reaches $S_{k,i}$.
- We start with $\lambda_{0,0} = 1$, and calculate $\lambda_{k,i}$ from $\lambda_{k-1,i}$ by induction:

$$\begin{cases} \lambda_{k,0} = e^{-r\Delta t}(p_{k,0}\lambda_{k-1,0}) \\ \lambda_{k,i} = e^{-r\Delta t}((1-p_{k,i-1})\lambda_{k-1,i-1} + p_{k,i}\lambda_{k-1,i}), \quad 1 \le i \le k-1 \\ \lambda_{k,k} = e^{-r\Delta t}\lambda_{k-1,k-1}(1-p_{k,k-1}). \end{cases} \tag{3}$$

Matching Option Prices

Under the binomial tree's discrete distribution, the option prices $C(K, k\Delta t)$ and $P(K, k\Delta t)$ are:

$$C(K, k\Delta t) = \sum_{i=0}^{k} \lambda_{k,i} \max(S_{k,i} - K, 0) \tag{4}$$

$$P(K, k\Delta t) = \sum_{i=0}^{k} \lambda_{k,i} \max(K - S_{k,i}, 0) \tag{5}$$

- Note that we have Condition 1 matching the forward prices, due to call-put parity, if call prices match, put prices will match by definition.
- We match call price on the upper part of the tree, and match put price on the lower part of the tree
- Now, option prices give us k constraitns, forward prices give us k constraints, we have $2k+1$ variables — left with 1 degree of freedom. We use that to pick the middle point(s).

When k is Even

At the even time steps, there are odd number of nodes $(k+1)$. We set the center point to be equal to S_0:

$$S_{k,k/2} = S_0 \tag{6}$$

The call prices above the central node, $C(S_{k-1,i}, k\Delta t)$, for $i < k/2$, can be expressed as:

$$C(S_{k-1,i}, k\Delta t) = \sum_{j=0}^{i} \lambda_{k,j}(S_{k,j} - S_{k-1,j}) = e^{-r\Delta t}[\underbrace{p_{k,0}\lambda_{k-1,0}}_{e^{r\Delta t}\lambda_{k,1}}(S_{k,0} - S_{k-1,i}) \tag{7}$$

$$+ \underbrace{((1-p_{k,0})\lambda_{k-1,0} + p_{k,1}\lambda_{k-1,1})}_{e^{r\Delta t}\lambda_{k,2}}(S_{k,1} - S_{k-1,i}) + \underbrace{((1-p_{k,1})\lambda_{k-1,1} + p_{k,2}\lambda_{k-1,2})}_{e^{r\Delta t}\lambda_{k,0}}(S_{k,2} - S_{k-1,i}) \tag{8}$$

$$+ \ldots + \underbrace{((1-p_{k,i-1})\lambda_{k-1,i-1} + p_{k,i}\lambda_{k-1,i})}_{e^{r\Delta t}\lambda_{k,i}}(S_{k,i} - S_{k-1,i})] \tag{9}$$

$$= e^{-r\Delta t}[\lambda_{k-1,0}\underbrace{(p_{k,0}(S_{k,0} - S_{k-1,i}) + p_{k,1}(S_{k,1} - S_{k-1,i}))}_{F_{k-1,0} - S_{k-1,i}} + \lambda_{k-1,1}(F_{k-1,1} - S_{k-1,i}) \tag{10}$$

$$+ \ldots + \lambda_{k-1,i-1}(F_{k-1,i-1} - S_{k-1,i}) + p_{k,i}\lambda_{k-1,i}(S_{k,i} - S_{k-1,i})] \tag{11}$$

$$= e^{-r\Delta t}\left[\underbrace{\sum_{j=0}^{i-1}\lambda_{k-1,j}(F_{k-1,j} - S_{k-1,i})}_{\text{known quantities, denoted as } \sum} + \underbrace{p_{k,i}}_{\frac{F_{k-1,i} - S_{k,i+1}}{S_{k,i} - S_{k,i+1}}} \lambda_{k-1,i}(S_{k,i} - S_{k-1,i})\right] \tag{12}$$

Re-arranging the formula, we obtain $S_{k,i}$ expressed in $S_{k,i+1}$:

$$S_{k,i} = \frac{S_{k,i+1}(e^{r\Delta t}C(S_{k-1,i}, k\Delta t) - \sum) - \lambda_{k-1,i}S_{k-1,i}(F_{k-1,i} - S_{k,i+1})}{e^{r\Delta t}C(S_{k-1,i}, k\Delta t) - \sum - \lambda_{k-1,i}(F_{k-1,i} - S_{k,S_i+1})} \tag{13}$$

This can be used to calculate the position of nodes above the middle point: $S_{k,i}$ for $i < k/2$. Similarly, using the put option prices, we can express $S_{k,i+1}$ in $S_{k,i}$:

$$S_{k,i+1} = \frac{S_{k,i}(e^{r\Delta t}P(S_{k-1,i}, k\Delta t) - \sum) + \lambda_{k-1,i}S_{k-1,i}(F_{k-1,i} - S_{k,i})}{e^{r\Delta t}P(S_{k-1,i}, k\Delta t) - \sum + \lambda_{k-1,i}(F_{k-1,i} - S_{k,S_i})} \tag{14}$$

Algorithm for even number of time steps:

- Start from center point $S_{k,k/2} = S_0$
- For nodes above center, calculate their positions iteratively using (13)
- For nodes below center, calculate their positions iteratively using (14)
- Calcualte the probability using (2)
- Note that in [Derman and Kani, 1994], the call and put prices $C(.)$ and $P(.)$ are calculted from a standard CRR binomial tree with interpolated smile. This is unecessary. We can just apply Black-Scholes formulae using the implied volatility queried from the implied volatility surface.

When k is Odd

We impose the constraint $S_0^2 = S_{k,\lfloor k/2 \rfloor} \times S_{k,\lfloor k/2 \rfloor+1}$ similar to CRR binomial tree. Substituting $S_{k,\lfloor k/2 \rfloor} = S_0^2 / S_{k,\lfloor k/2 \rfloor+1}$ into (13), we can obtain:

$$S_{k,\lfloor k/2 \rfloor} = \frac{S_0[e^{r\Delta t} C(S_0, k\Delta t) + \lambda_{k-1,\lfloor k/2 \rfloor} S_0 - \sum]}{\lambda_{k-1,\lfloor k/2 \rfloor} F_{k-1,\lfloor k/2 \rfloor} - e^{r\Delta t} C(S_0, k\Delta t) + \sum} \tag{15}$$

Once we position $S_{k,\lfloor k/2 \rfloor}$ and $S_{k,\lfloor k/2 \rfloor+1}$, we can use the same iterative method as when k is even, to setup the rest of the tree nodes.

Working examples are available in [Derman and Kani, 1994], node index there is different from our notation: we are indexing from top to bottom — $S_{k,0}$ represents the top node at time step k, in the paper the index is from bottom to top.

Project Requirements

- Implement the above implied binomial tree model and pricer. For discussion simplicity, we ommited the dividend rate q. In the actual implementation, we need to incorporate dividend rate q.
- Test the tree with different tradeables:
    - Construct test cases to demonstrate that the model is implemented correctly.
    - Discuss observations from the test cases. Suggest the appropriate model settings (discretization steps).
- Project group: one or two members
- A project report together with source code to be submitted, in pdf format.
- Assessment will be made on the overall quality of report, code and result
- **Deadline: 19 Apr 2020, 23:59:59**

References

Derman, E. and Kani, I. (1994).
The volatility smile and its implied tree
http://www.cmat.edu.uy/~mordecki/hk/derman-kani.pdf.
Quantitative Strategies Research Notes.