Lecture 15 (10/6). Please enable video if you can.

6.2. **State processes.**

Question 6.14. *Consider the N-period binomial model, and a security with payoff V_N. Let X_n be the arbitrage free price at time $n \leqslant N$, and Δ_n be the number of shares in the replicating portfolio. What is an algorithm to find X_n, Δ_n for all $n \leqslant N$? How much is the computational time?*

X_n = AFP at time n = Wealth of Rep Port at time n

$$= \frac{1}{D_n} \tilde{E}_n (D_N V_N) = \frac{1}{(1+r)^{N-n}} \tilde{E}_n V_N \quad \left(\text{if } D_n = \frac{1}{(1+r)^n} \right)$$

$\Rightarrow$ Say maturity N = 3 months $\longrightarrow$ 90 days.

Computational cost $\simeq O(\#\text{ elmts in } \Omega) = O(2^N)$

$$= O(2^{90})$$

$$2^{10} = 1024 \approx 10^3 \Rightarrow 2^{90} = (2^{10})^9 \approx 10^{27}$$

Say can compute $O(10^9)$ operations in one second.

Computational time $\approx O(10^{18})$ seconds.

Life of universe $\approx 10^{19}$ seconds!!

Theorem 6.15. *Suppose a security pays $V_N = g(S_N)$ at maturity N for some (non-random) function g. Then the arbitrage free price at time $n \leqslant N$ is given by $V_n = f_n(S_n)$, where:*

(1) $f_N(x) = g(x)$ *for* $x \in \mathrm{Range}(S_N)$.

(2) $f_n(x) = \dfrac{1}{1+r}(\tilde{p} f_{n+1}(ux) + \tilde{q} f_{n+1}(dx))$ *for* $x \in \mathrm{Range}(S_n)$.

Remark 6.16. Reduces the computational time from $O(2^N)$ to $O(\sum_0^N |\mathrm{Range}(S_n)|) = O(N^2)$ for the Binomial model.

Remark 6.17. Can solve this to get $f_n(x) = \dfrac{1}{(1+r)^{N-n}} \displaystyle\sum_{k=0}^{N-n} \binom{N-n}{k} \tilde{p}^k \tilde{q}^{N-n-k} f_N(x u^k d^{N-n-k})$

Note: Range $S_1 = \{uS_0, dS_0\} \Rightarrow \#\,\mathrm{Range}(S_1) = 2$.

$\mathrm{Range}(S_2) = \{u^2 S_0, udS_0, d^2 S_0\} \Rightarrow \#\,\mathrm{Range}(S_2) = 3$

$\vdots$

$\mathrm{Range}(S_n) = \{u^n S_0, u^{n-1} d S_0, \dots, d^n S_0\}$, $\#$ elem $\mathrm{Range}(S_n) = n$.

① Know $f_N(x) = g(x)$ $\qquad x \in \text{Range}(S_N)$

② $$f_{N-1}(x) = \frac{1}{1+r}\Big(f_N(ux)\,\tilde{p} + f_N(dx)\,\tilde{q}\Big) \qquad x \in \text{Range}(S_{N-1})$$

③ $$f_{N-2}(x) = \frac{1}{1+r}\Big(f_{N-1}(ux)\,\tilde{p} + \underbrace{f_{N-1}(dx)\,\tilde{q}}_{}\Big)$$

$$= \frac{1}{1+r}\left(\frac{1}{1+r}\Big(f_N(u^2x)\,\tilde{p} + f_N(udx)\,\tilde{q}\Big)\tilde{p} + \frac{1}{1+r}\Big(f_N(udx)\,\tilde{p} + f_N(d^2x)\,\tilde{q}\Big)\tilde{q}\right)$$

$$= \frac{1}{(1+r)^2}\left(f_N(u^2 x)\tilde{p}^2 + 2\tilde{p}\tilde{q}\, f_N(udx) + f_N(d^2x)\tilde{q}^2 \right)$$

Pf of Thm 6.15. Backward induction.

① Let $f_N(x) = g(x)$. $\Rightarrow$ $f_N(S_N) = g(S_N) = V_N$.

② Let $n = N-1$. NTS AFP at time n is $f_n(S_n)$

where $f_n(x) = \frac{1}{1+r}\left(f_{n+1}(ux)\tilde{p} + f_{n+1}(dx)\tilde{q} \right)$

Let X_n = AFP at time n.

Know $$X_n = \frac{1}{D_n}\tilde{E}_n\left(D_N V_N\right)$$

Since $n = N-1$ $\Rightarrow$ $$X_n = \frac{1}{1+r}\tilde{E}_n\left(f_{n+1}\left(S_{n+1}\right)\right)$$

$$= \frac{1}{1+r}\tilde{E}_n\left(\underbrace{f_{n+1}}_{\text{not random}}\left(\underbrace{S_n}_{\mathcal{F}_n\text{-meas}}\underbrace{Y_{n+1}}_{\mathcal{F}_n\text{-ind}}\right)\right) \quad \text{where } Y_{n+1} = \begin{cases} u & \omega_{n+1}=1 \\ d & \omega_{n+1}=-1 \end{cases}$$

$$\overset{\text{independence lemma}}{=} \frac{1}{1+r}\left(\tilde{p}\, f_{n+1}(S_n \cdot u) + \tilde{q}\, f_{n+1}(S_n d)\right)$$

$$\therefore \text{ AFP at time } n = X_n = \underbrace{\frac{1}{1+r}\left(\tilde{p}\, f_{n+1}(S_n u) + \tilde{q}\, f_{n+1}(S_n d)\right)}_{\text{is a fn of } S_n.}$$

Let $f_n(x) = \frac{1}{1+r}\left(\tilde{p}\, f_{n+1}(x u) + \tilde{q}\, f_{n+1}(x d)\right) \Rightarrow X_n = f_n(S_n)$

u

QED.

Above was for $n = N-1$.

In general: Backward induction

① Suppose $X_{n+1} = $ AFP at time $n+1 = f_{n+1}(S_{n+1})$.

② AFP at time n:

$$X_n = \frac{1}{D_n}\tilde{E}_n(D_N V_N)$$

$$= \frac{1}{D_n}\tilde{E}_n \underbrace{\tilde{E}_{n+1}(D_N V_N)}$$

$$= \frac{1}{D_n}\tilde{E}_n(D_{n+1} X_{n+1}) = \frac{1}{1+r}\tilde{E}_n f_{n+1}(S_{n+1})$$

(same reason as before) $= \frac{1}{1+r}\left(f_{n+1}(u S_n)\,\tilde{p} + f_{n+1}(d S_n)\,\tilde{q} \right)$

$$= f_n(S_n), \quad \text{where } f_n(x) = \frac{1}{1+r}\left[f_{n+1}(ux)\,\tilde{p} + f_{n+1}(dx)\,\tilde{q} \right]$$

QED

Note: Above algorithm works to price any security that is a fn of the stock price at maturity (e.g. call/put options)

Question 6.18. *How do we handle other securities? E.g. Asian options (of the form $g(\sum_0^N S_k)$)?*

Eg: Asian call option strike K & maturity N.

pays $$\left[\left(\frac{1}{N+1}\sum_{n=0}^{N} S_n\right) - K\right]^+$$

Can't directly use above alg to price Asian options

but: Can if we "expand the state process."

↳ add

Definition 6.24. We say a d-dimensional process $Y = (Y^1, \dots, Y^d)$ process is a *state process* if for any security with maturity $m \leqslant N$, and payoff of the form $V_m = f_m(Y_m)$ for some (non-random) function f_m, the arbitrage free price must also be of the form $V_n = f_n(Y_n)$ for some (non-random) function f_n.

Remark 6.25. For state processes given f_N, we find f_n by backward induction. The number of computations at time n is of order $\operatorname{Range}(Y_n)$.

Remark 6.26. The fact that S_n is Markov (under $\tilde{\boldsymbol{P}}$) implies that it is a state process.