Computational Methods in Finance, Lecture 1, Duality and Dynamic Programming.

Last revised September 1998

Jonathan Goodman *
Courant Institute of Mathematical Sciences, NYU

September 5, 1999

1 Introduction

Most valuation problems in quantitative finance represent the desired value as the expected value of some random variable together with a stochastic model of market movements.[1] Given the model, there are two ways to compute the expected value using evolution equations.[2] One method starts with expected values known in the future and computes expected values at successively earlier times until the present expected values are found. The other method starts with given probabilities for current market conditions and works forwards in time to find probabilities for market conditions at a desired future time. These two evolution equations are similar but not identical. One of the differences is the natural direction of time change, backwards for expected values and forwards for probabilities.

This lecture discusses these evolution equations in the simple case of discrete time and discrete "state space". The main ideas are contained here.

---

*goodman@cims.nyu.edu, or http://www.math.nyu.edu/faculty/goodman,

[1]This stochastic model is the "risk neutral equivalent martingale measure", not the historical evolution process.

[2]Valuation using Monte Carlo simulation will be covered later.

The extension to more complex situations, continuous time or state space, are mainly technical. The relationship between them is called "duality". It is an extension of the relationship between a matrix and its transpose.

Many financial instruments allow the holder to make decisions along the way that effect the ultimate value of the instrument. American style options, loans that be repaid early, and convertible bonds are examples. To compute the value of such an instrument, we also seek the optimal decision strategy. *Dynamic programming* is a computational method that computes the value and decision strategy at the same time. It reduces the difficult "multiperiod decision problem" to a sequence of hopefully easier "single period" problems. It works backwards in time much as the expectation method does. The tree method commonly used to value American style stock options is an example of the general dynamic programming method.

2 Markov Chains

(This section assumes familiarity with basic probability theory using mathematicians' terminology. References on this include the probability books by G. C. Rota, W. Feller, Hoel and Stone, and B. V. Gnedenko.)

Many discrete time discrete state space stochastic models are Markov chains. Such a Markov chain is characterized by its state space, $\mathcal{S}$, and its transition matrix, P. We use the following notations:

- x, y, $\ldots$: possible states of the system, elements of $\mathcal{S}$.
- The possible times are $t = 0, 1, 2, \ldots$.
- $X(t)$: the (unknown) state of the system at time t. It is some element of $\mathcal{S}$.
- $u(x,t) = \mathbf{Pr}(X(t) = x)$. These probabilities satisfy an evolution equation moving forward in time. We use similar notation for conditional probabilities, for example, $u(x,t|X(0) = x_0) = \mathbf{Pr}(X(t) = x|X(0) = x_0)$.
- $p(x,y) = \mathbf{Pr}(x \to y) = \mathbf{Pr}(X(t+1) = y|X(t) = x)$. These "transition probabilities" are the elements of the transition matrix, P.

The transition probabilities have the properties:

$$0 \leq p(x,y) \leq 1 \quad \text{for all } x \in \mathcal{S} \text{ and } y \in \mathcal{S}. \tag{1}$$

and

$$\sum_{y \in \mathcal{S}} p(x,y) = 1 \quad \text{for all } x \in \mathcal{S}. \tag{2}$$

The first is because the $p(x,y)$ are probabilities, the second because the state x must go somewhere, possibly back to x. It is not true that

$$\text{(NOT ALWAYS TRUE)} \qquad \sum_{x \in \mathcal{S}} p(x,y) = 1 \ . \qquad \text{(NOT ALWAYS TRUE)}$$

The Markov property is that knowledge of the state at time t is all the information about the present and past relevant to predicting the future. That is:

$$\mathbf{Pr}(X(t{+}1) = y | X(t) = x_0, X(t{-}1) = x_1, \ldots) = \mathbf{Pr}(X(t{+}1) = y | X(t) = x_0)$$

no matter what extra history information ($X(t-1) = x_1, \ldots$) we have. This may be thought of as a lack of long term memory. It may also be thought of as a completeness property of the model: the state space is rich enough to characterize the state of the system at time t completely.

The evolution equation for the probabilities $u(x,t)$ is found using conditional probability:

$$\begin{aligned} u(x,t+1) &= \mathbf{Pr}(X(t+1) = x) \\ &= \sum_{y \in \mathcal{S}} \mathbf{Pr}(X(t+1) = x | X(t) = y) \cdot \mathbf{Pr}(X(t) = y) \\ u(x,t+1) &= \sum_{y \in \mathcal{S}} p(y,x) u(y,t) \ . \end{aligned} \tag{3}$$

To express this in matrix form, we suppose that the state space, $\mathcal{S}$, is finite, and that the states have been numbered x_1, …, x_n. The transition matrix, P, is $n \times n$ and has (i,j) entry $p_{ij} = p(x_i, x_j)$. We sometimes conflate i with x_i and write $p_{xy} = p(x,y)$; until you start programming the computer, there is no need to order the states. With this convention, (3) can be interpreted as vector–matrix multiplication if we define a *row* vector $\underline{u}(t)$ with components $(u_1(t), \ldots, u_n(t))$, where we have written $u_i(t)$ for $u(x_i, t)$.

As long as ordering is unimportant, we could also write $u_x(t) = u(x,t)$. Now, (3) can be rewritten

$$\underline{u}(t+1) = \underline{u}(t)P \quad . \tag{4}$$

Since $\underline{u}$ is a row vector, the expression $P\underline{u}$ does not make sense because the dimensions of the matrices are incompatible for matrix multiplication. The convention of using a row vector for the probabilities and therefore putting the vector in the left of the matrix is common in applied probability. The relation (4) can be used repeatedly to yield

$$\underline{u}(t) = \underline{u}(0)P^t \quad , \tag{5}$$

where P^t means P to the power t, not the transpose of P.

3 Expected Values

There are several situations in which expected (present values of) payouts can be computed using an evolution equation that has time moving backwards from the future to the present. The basic idea comes through clearly in the simple case of an undiscounted terminal payout. At the terminal time, T, we get a payout that depends on the state of the system at that time: $f_T(X(T))$. We want to compute the expected value of this payout:

$$\mathbf{E}\left[f_T(X(T))\right] \quad . \tag{6}$$

To compute this, we compute a connected collection of expectation values, $f(x,t)$, defined as

$$f(x,t) = \mathbf{E}\left[f_T(X(T)) | X(t) = x\right] \quad . \tag{7}$$

We find a relationship between these numbers by considering one step of the Markov chain. If the system is in state x at time t, then the probability for it to be at state y at the next time is $p(x \to y) = p(x,y)$. For expectation values, this implies

$$\begin{aligned} f(x,t) &= \mathbf{E}\left[f_T(X(T)) | X(t) = x\right] \\ &= \sum_{y \in \mathcal{S}} \mathbf{E}\left[f_T(X(T)) | X(t+1) = y\right] \cdot \mathbf{Pr}\left(X(t+1) = y \mid X(t) = x\right) \\ f(x,t) &= \sum_{y \in \mathcal{S}} f(y,t+1) p(x,y) \quad . \end{aligned} \tag{8}$$

This relation is used to compute (5) as follows. The final time values, $f(x,T)$ are the given values $f_T(x)$. From these, we compute all the numbers $f(x,T-1)$ using (7) with $t=T-1$. Continuing like this, we eventually get to $t=0$. We may know $X(0)$, the state of the system at the current time. Otherwise we can use

$$\begin{aligned}\mathbf{E}\left[f_T(X(T))\right] &= \sum_{x\in\mathcal{S}} \mathbf{E}\left[f_T(X(T))|X(0)=x\right]\cdot \mathbf{Pr}\left(X(0)=x\right)\\ &= \sum_{x\in\mathcal{S}} f(x,0)u(x,0) \ .\end{aligned}$$

All the values on the bottom line should be known.

As with the probability evolution equation (3), the equation for the evolution of the expectation values (8) can be written in matrix form. The difference from the probability evolution equation is that here we arrange the numbers $f_j = f(x_j,t)$ into a *column* vector, $\underline{f(t)}$. The evolution equation for the expectation values is then written in matrix form as

$$\underline{f}(t) = P\underline{f}(t+1) \ . \tag{9}$$

This time, the vector goes on the right. If apply (**??**) repeatedly, we get, in place of (5),

$$\underline{f}(t) = P^{T-t}\underline{f}(T) \ . \tag{10}$$

There are several useful variations on this theme. For example, suppose that we have a running payout rather than a final time payout. Call this payout $g(x,t)$. If $X(t)=x$ then $g(x,t)$ is added to the total payout that accumulates over time from $t=0$ to $t=T$. We want to compute

$$\mathbf{E}\left[\sum_{t=0}^{T} g(X(t),t)\right] \ .$$

As before, we find this by computing more specific expected values:

$$f(x,t) = \mathbf{E}\left[\sum_{t'=t}^{T} g(X(t'),t')|X(t)=x\right] \ .$$

These numbers are related through a generalization of (7) that takes into account the known contribution to the sum from the state at time t:

$$f(x,t) = \sum_{y\in\mathcal{S}} f(y,t+1)p(x,y) + g(x,t) \ .$$

The "initial condition", given at the final time, is

$$f(x,T) = g(x,T) \quad .$$

This includes the previous case, we take $g(x,T) = f_T(x)$ and $g(x,t) = 0$ for $t < T$.

As a final example, consider a path dependent discounting. Suppose for a state x at time t there is a discount factor $r(x,t)$ in the range $0 \leq r(x,t) \leq 1$. A cash flow worth f at time $t+1$ will be worth $r(x,t)f$ at time t if $X(t) = x$. We want the discounted value at time $t = 0$ at state $X(0) = x$ of a final time payout worth $f_T(X(T))$ at time T. Define $f(x,t)$ to be the value at time t of this payout, given that $X(t) = x$. If $X(t) = x$ then the time $t+1$ expected discounted (to time $t+1$) value is

$$\sum_{y \in \mathcal{S}} f(y, t+1) p(x,y) \quad .$$

This must be discounted to get the time t value, the result being

$$f(x,t) = r(x,t) \sum_{y \in \mathcal{S}} f(y, t+1) p(x,y) \quad .$$

4 Duality and Qualitative Properties

The forward evolution equation (**??**) and the backward equation (8) are connected through a duality relation. For any time t, we compute (7) as

$$\begin{aligned} \mathbf{E}\left[f_T(X(T))\right] &= \sum_{x \in \mathcal{S}} \mathbf{E}\left[f_T(X(T)) | X(t) = x\right] \cdot \mathbf{Pr}(X(t) = x) \\ &= \sum_{x \in \mathcal{S}} f(x,t) u(x,t) \quad . \end{aligned} \tag{11}$$

For now, the main point is that the sum on the bottom line does not depend on t. Given the constancy of this sum and the u evolution equation (4), we can give another derivation of the f evolution equation (7). Start with

$$\sum_{x \in \mathcal{S}} f(x, t+1) u(x, t+1) = \sum_{y \in \mathcal{S}} f(y,t) u(y,t) \quad .$$

Then use (4) on the left side and rearrange the sum:

$$\sum_{y\in\mathcal{S}}\left(\sum_{x\in\mathcal{S}} f(x,t+1)p(y,x)\right)u(y,t) = \sum_{y\in\mathcal{S}} f(y,t)u(y,t) \ .$$

Now, if this is going to be true for any $u(y,t)$, the coefficients of $u(y,t)$ on the left and right sides must be equal for each y. This gives (7). Similarly, it is possible to derive (4) from (7) and the constancy of the expected value.

The evolution equations (4) and (7) have some qualitative properties in common. The main one being that they preserve positivity. If $u(x,t)\geq 0$ for all $x\in\mathcal{S}$, then $u(x,t+1)\geq 0$ for all $x\in\mathcal{S}$ also. Likewise, if $f(x,t+1)\geq 0$ for all x, then $f(x,t)\geq 0$ for all x. These properties are simple consequences of (4) and (7) and the positivity of the $p(x,y)$. Positivity preservation does not work in reverse. It is possible, for example, that $f(x,t+1)<0$ for some x even though $f(x,t)\geq 0$ for all x.

The probability evolution equation (4) has a conservation law not shared by (7). It is

$$\sum_{x\in\mathcal{S}} u(x,t) = const\,. \tag{12}$$

independent of t. This is natural if u is a probability distribution, so that the constant is 1. The expected value evolution equation (7) has a *maximum principle*

$$\max_{x\in\mathcal{S}} f(x,t) \leq \max_{x\in\mathcal{S}} f(x,t+1) \ . \tag{13}$$

This is a natural consequence of the interpretation of f as an expectation value. The probabilities, $u(x,t)$ need not satisfy a maximum principle either forward of backward in time.

This duality relation has is particularly transparent in matrix terms. The formula (7) is expressed explicitly in terms of the probabilities at time t as

$$\sum_{x\in\mathcal{S}} f(x,T)u(x,T) \ ,$$

which has the matrix form

$$\underline{u}(t)\underline{f}(T) \ .$$

Written in this order, the matrix multiplication is compatible; the other order, $\underline{f}(T)\underline{u}(T)$, would represent an $n\times n$ matrix instead of a single number. In view of (**??**), we may rewrite this as

$$\underline{u}(0)P^t\underline{f}(T) \ .$$

Because matrix multiplication is associative, this may be rewritten

$$\left[\underline{u}(0)P^t\right] \cdot \left[P^{T-t}\underline{f}(T)\right] \tag{14}$$

for any t. This is the same as saying that $\underline{u}(t)\underline{f}(T-t)$ is independent of t, as we already saw.

In linear algebra and functional analysis, "adjoint" or "dual" is a fancy generalization of the transpose operation of matrices. People who don't like to think of putting the vector to the left of the matrix think of $\underline{u}P$ as multiplication of (the transpose of) $\underline{u}$, on the right, by the transpose (or adjoint or dual) of P. In other words, we can do enough evolution to compute an expected value either using P its dual (or adjoint or transpose). This is the origin of the term "duality" in this context.

5 Dynamic Programming

Dynamic programming is a method for valuing American style options and other financial instruments that allow the holder to make decisions that effect the ultimate payout. The idea is to define the appropriate value function, $f(x,t)$, that satisfies a nonlinear version of the backwards evolution equation (7). I will explain the idea in a simple but somewhat abstract situation. From in the previous section, it is possible to use these ideas to treat other related problems.

We have a Markov chain as before, but now the transition probabilities depend on a "control parameter", ξ. That is

$$p(x,y,\xi) = \mathbf{Pr}\left(X(t+1) = y | X(t) = x, \xi\right) \quad .$$

In the "stochastic control problem", we are allowed to choose the control parameter at time t, $\xi(t)$, knowing the value of $X(t)$ but not any more about the future than the transition probabilities. Because the system is a Markov chain, knowledge of earlier values, $X(t-1), \ldots$, will not help predict or control the future. Choosing ξ as a function of $X(t)$ and t is called "feedback control" or a "decision strategy". The point here is that the optimal control policy is a feedback control. That is, instead of trying to choose a whole control trajectory, $\xi(t)$ for $t = 0, 1, \ldots, T$, we instead try to choose the feedback functions $\xi(X(t), t)$. We will write $\xi(X,t)$ for such a decision strategy.

Any given strategy has an expected payout, which we write

$$\mathbf{E}_\xi \left[f_T(X(T)) \right] \quad .$$

Our object is to compute the value of the financial instrument under the optimal decision strategy:

$$\max_\xi \mathbf{E}_\xi \left[f_T(X(T)) \right] \quad , \tag{15}$$

and the optimal strategy that achieves this.

The appropriate collection of values for this is the "cost to go" function

$$f(x,t) = \max_\xi \mathbf{E}_\xi \left[f_T(X(T)) | X(t) = x \right] \quad . \tag{16}$$

As before, we have "initial data" $f(x,T) = f_T(x)$. We need to compute the values $f(x,t)$ in terms of already computed values $f(x,t+1)$. For this, we suppose that the optimal decision strategy at time t is not yet known but those at later times are already computed. If we use control variable $\xi(t)$ at time t, and the optimal control thereafter, we get payout depending on the state at time $t+1$:

$$\mathbf{E} \left[f(X(t+1), t+1) | X(t) = x, \xi(t) \right] = \sum_{y \in \mathcal{S}} f(y, t+1) p(x, y, \xi(t)) \quad .$$

Maximizing this expected payout over $\xi(t)$ gives the optimal expected payout at time t:

$$f(x,t) = \max_{\xi(t)} \sum_{y \in \mathcal{S}} f(y, t+1) p(x, y, \xi(t)) \quad . \tag{17}$$

This is the principle of dynamic programming. We replace the "multiperiod optimization problem" (11) with a sequence of hopefully simpler "single period" optimization problems (13) for the cost to go function.