January 24, 2005

Lecture 2 Recursive sequences, linearity, and some applications.

Consider given unknown quantities $a_1, a_2, \ldots, a_n, \ldots$ A *recursive formula* is a formula which relates the general term a_n to some of its predecessors. The sequence is called recursive if it obeys the formula.

Example 1. Let $a_n = a_{n-1}a_{n-2}$, that is, a term is the product of its two immediate predecessors. A possible such sequence would be $a_1 = 1$, $a_2 = 2$, $a_3 = 2$, $a_4 = 4$, $a_5 = 8$, $a_6 = 32$, etc. Another such sequence would be $b_1 = 3$, $b_2 = 2$, $b_3 = 6$, $b_4 = 12$, etc. Notice that the sequence is completely determined by its first two terms, but these two may take any value we want. The free initial terms of a recursive sequence are called *seeds*. Here's another example.

Example 2. Suppose the annual interest rate at a bank is 6%, compounded monthly. This means that if you have A dollars in the bank today, then next month you will have

$$A\left(1 + \frac{.06}{12}\right)$$

in the bank. If we call A_n the quantity in the bank at month n, then

$$A_n = A_{n-1}\left(1 + \frac{.06}{12}\right) = A_{n-1}r,$$

where A_1 is the quantity you started with, and r is the multiplier. Here is a more complicated example.

Milk example. There are three brands of milk, A, B, and C, and consumer research reveals the following data: from month to month

- 60% of consumers of A change to B, and 20% change to C;
- 35% of consumers of B change to A, and 40% to C;
- 25% of consumers of C change to A, and 20% to B.

If we denote by a_n, b_n, and c_n the population drinking brands A, B, and C respectively during month n, then we have the following recursive system

$$\begin{aligned}
a_{n+1} &= .20a_n + .35b_n + .25c_n \\
b_{n+1} &= .60a_n + .25b_n + .20c_n \\
c_{n+1} &= .20a_n + .40b_n + .55c_n
\end{aligned}$$

Even though a_{n+1} depends not only on a_n, but also on the other sequences, we still call this a recursive sequence, because we can view *the whole data a_n, b_n, and c_n as a single state of the world.* With that in mind, the state of the world this month depends only on what it was last month.

A recursive formula is called *linear* if it behaves well with respect to addition and multiplication by a constant. What this means is that if two sequences a_n and b_n satisfy the same recursive formula, then the new sequence c_n given by the sum $c_n = a_n + b_n$ will also satisfy the same recursive formula. Also, if d is a number, then the sequence $c_n = da_n$ satisfies the same recursive formula.

Our first example is *not* linear, because if we put $c_1 = a_1 + b_1$, $c_2 = a_2 + b_2$, etc., then $c_1 = 1 + 2 = 3$, $c_2 = 3 + 2 = 5$, $c_3 = 2 + 6 = 8$, but we immediately see that we don't have $c_3 = c_1 c_2$, and so the sequence c_n does not satisfy the same recursive formula.

Our second example is linear. Say that $A_{n+1} = A_n r$, and $B_{n+1} = B_n r$. Put $C_n = A_n + B_n$. Then

$$C_{n+1} = A_{n+1} + B_{n+1} = A_n r + B_n r = (A_n + B_n)r = C_n r,$$

and also if d is any number, and we put $C_n = dA_n$, then

$$C_{n+1} = dA_{n+1} = d(A_n r) = (dA_n)r = C_n r.$$

Linearity is a good thing. Most of the equations dealing with finance are linear. What it means is that if you invest twice the amount, then your returns are doubled (that is the second condition), and if your individual investments give you individual returns, then your total return is the sum of the individual parts (the first condition).

Is the milk example linear? That seems to be much more difficult to verify, since we need the state comprising a_n, b_n, and c_n, and then a second state comprising d_n, e_n, and f_n, and then we need to sum those states respecting the order in which they appear... Very messy. To make things easier to write and understand, we borrow language from linear algebra. We put the states a_n, b_n, and c_n in an ordered array, called a *vector*, and write it vertically, and name it v_n, like this

$$v_n = \begin{pmatrix} a_n \\ b_n \\ c_n \end{pmatrix}.$$

The percentages showing up in the milk problem, we write in an array which we call a *matrix array*, or simply *matrix*. We name this matrix M.

$$M = \begin{pmatrix} .20 & .35 & .25 \\ .60 & .25 & .20 \\ .20 & .40 & .55 \end{pmatrix}.$$

The formulas defining the milk problem can now be written in symbolic form as

$$v_{n+1} = Mv_n.$$

This is what the symbol Mv_n means:

$$\begin{pmatrix} a_{n+1} \\ b_{n+1} \\ c_{n+1} \end{pmatrix} = Mv_n = \begin{pmatrix} .20 & .35 & .25 \\ .60 & .25 & .20 \\ .20 & .40 & .55 \end{pmatrix} \begin{pmatrix} a_n \\ b_n \\ c_n \end{pmatrix} = \begin{pmatrix} .20a_n + .35b_n + .25c_n \\ .60a_n + .25b_n + .20c_n \\ .20a_n + .40b_n + .55c_n \end{pmatrix}$$

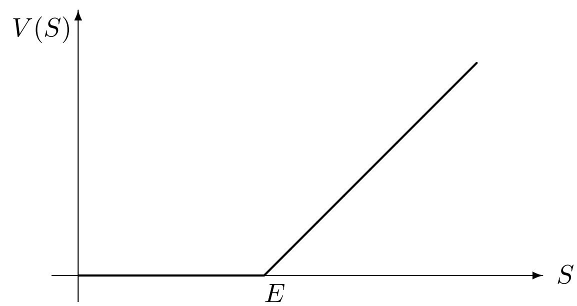
We can view the last equality as the definition of the effect of a matrix on a vector, or the definition of a matrix applied to a vector.

Financial interlude. Say you bought a call option with a strike price E and expiry date T. Let S be the value of the underlying asset *at the expiry date*. The question is: how much is the option worth for you at the expiry date T?

Well, if $S < E$ at the expiry date, this means the asset is not worth the strike price, and of course you won't exercise the option for the strike price E since you can right now buy the asset more cheaply somewhere else. So the option is worthless, with a value of zero. On the other hand, if $S > E$, then you exercise your option and immediately sell the asset. Your profit is $S - E$, and that is the option's value. In any case, the value $V(S)$ (depending on S) is

$$V(S) = \max\{0, S - E\}.$$

Here's a graph of what this looks like.



This graph is called a *payoff diagram.* As an exercise draw the payoff diagram of a put option. *It is important to notice that the payoff diagram does not take into account the original value of the option!* This is a matter of convention.

Back to Math. The recurrence given by the matrix equation described before is linear. Even thought there are many symbols, one can verify that if v and w are vectors, d is a number, and M is a matrix, then

$$M(v + w) = Mv + Mw, \qquad \text{and} \qquad M(dv) = d(Mv).$$

Matrices don't have to be square arrays. For example

$$M = \begin{pmatrix} 1 & 3 & -1 \\ 4 & 0 & 1 \end{pmatrix}$$

is a perfectly good matrix of type "two by three", or 2x3. The first number is the number of rows, the second is the number of columns. If we think of vectors in this way, then a vector is a matrix of type mx1, where m is the number of entries. The way to multiply matrices by vectors is illustrated in the example

$$\begin{pmatrix} 1 & 3 & -1 \\ 4 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \cdot a + 3 \cdot b + (-1) \cdot c \\ 4 \cdot a + 0 \cdot b + 1 \cdot c \end{pmatrix}$$

Please notice: a 2x3 matrix transforms a 3x1 vector into a 2x1 vector. In general an mxn matrix transforms an nx1 vector into an mx1 vector.

Let's present some applications of the concepts we saw.

Application 1 (a fake application): the normal distribution. We know that the function e^{-t^2} does not have a closed form integral, that is

$$F(x) = \int_0^x e^{-t^2}\,dt$$

can't be written in terms of simple functions. And yet the values of $F(x)$ are very important in probability and statistics. Here is a way to produce approximate values for $F(.01)$, $F(0.2)$, $F(.03)$, etc. By the fundamental theorem of Calculus we know that $F'(x) = e^{-x^2}$; on the other hand (by linear approximation)

$$F(x + .01) \approx F(x) + .01\,e^{-x^2}.$$

If we now set $x_0 = 0$, $x_1 = .01$, $x_2 = .02$, $\ldots$, $x_n = .01\,n$, we can define the recursive sequence

$$f_0 = 0, \qquad\qquad f_{n+1} = f_n + .01\,e^{-x_n^2}.$$

Notice how this sequence is no more than the area of the internal rectangles with base length .01. In this example f_n is an approximation of the value $F(x_n)$. This is a fake application because the area under the Gaussian is not computed in this way.

Application 2: the milk example. If you are a producer of milk brand A, B, or C, one question you would very much like to see answered is: am I going out of business? We can answer part of this question right away, because if (say) A runs out of customers this month, then next month it will receive quite a few patrons from B and C, so as long as A can hang on for a while, customers will come back to A. That is not a very good answer, because in practice for a business to survive it needs a minimal amount of customers, so a better question is: knowing that A needs at least p customers per month to stay in business, can we predict the future of A? Say, what happens if $p = 30$?

In general, can we predict the future of the recursive sequence? For instance, will it stabilize?

We will not answer the question of whether the sequence stabilizes or not, but *if it does stabilize*, then the populations consuming A, B, and C become numerically stable from month to month. Call these populations a, b, and c. We must have that

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} .20 & .35 & .25 \\ .60 & .25 & 20 \\ .20 & .40 & .55 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Let's say that the total population is 100 people. We can solve the equations and find

$$a = \frac{10,300}{383} \approx 27 \qquad\qquad b = \frac{12,400}{383} \approx 32 \qquad\qquad c = \frac{15,600}{383} \approx 41$$

These fractions should be rounded to appropriate integers, but they answer the question about the survival of the three brands–as long as the recursive sequence tends to a limit. By the way, if $p = 30$, then A should be very worried about the business.

Problems.

1. In this problem we will see that a certain recursive sequence converges to the value $\sqrt{2}$. Given the recursive formula and seed

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{2}{x_n}\right), \qquad\qquad x_0 = 1,$$

a) find x_2, x_3, and x_4 (you may use a calculator if you want);
b) show that *if* $x_n \to L$, then $L = \sqrt{2}$;
c) show that for all $x > 0$ we have $\sqrt{2} \leq \frac{1}{2}(x + \frac{2}{x})$;
d) show that if $x > \sqrt{2}$, then $\frac{1}{2}(x + \frac{2}{x}) < x$, and conclude that

$$x_n \to \sqrt{2}.$$

2. Given a 3x3 matrix M whose individual rows each add up to 1 find a 3x1 vector v (not all zero) such that $v = Mv$. (Hint: Do a few examples.)

3. Consider 5 adjacent nodes labeled a_1, a_2, $\ldots a_5$ as in the picture.

5	-2	7	4	3
•	•	•	•	•
a_1	a_2	a_3	a_4	a_5

A value was assigned to each node (the value of a_1 is 5, etc). This is the starting configuration (or seed, at time $t = 0$). Now we change the configuration obeying the following rule: the values of a_1 and a_5 remain the same, but every other node takes the value given by the *average of the values of the two adjacent nodes.* The new configuration is our time $t = 1$. We repeat this process for times 2, 3, etc. Denote by $a_j(n)$ the value of node j at time n.

a) Find recursive formulas for $a_2(n+1)$, $a_3(n+1)$, and $a_4(n+1)$.
b) Let v_{n+1} be the vector $\begin{pmatrix} a_2(n+1) \\ a_3(n+1) \\ a_4(n+1) \end{pmatrix}$. Find the 3x3 matrix M such that

$$v_{n+1} = Mv_n + \begin{pmatrix} 5/2 \\ 0 \\ 3/2 \end{pmatrix}.$$

c) If the process converges, what is the limit?
d) Is the recursive formula linear? Why?

4. Modify the previous problem in the following way: now the values of the endpoints a_1 and a_5 change to new values given by their average with the nearest neighbor. Use the same seeds as before.

a) Find recursive formulas for $a_1(n+1)$,..., $a_5(n+1)$.

b) Show that $a_1(n+1) + a_2(n+1) + a_3(n+1) + a_4(n+1) + a_5(n+1) = 17$, for all n.

b) Find the 5x5 matrix M such that $v_{n+1} = Mv_n$ (here v_n has 5 components).

c) Find the limit if it exists. (Hint: exercise 3, and item b above.)

5. Draw the payoff diagram for a put option with strike price E.

6. Draw the payoff diagram for the following portfolio: you acquire a put option P and a call option C, both with the same expiry date and strike price E.