Kansas State University | Department of Mathematics

Applied Matrix Theory - Math 551

Discrete Dynamical Systems

Created by Prof. Diego Maldonado and Prof. Virginia Naibo

In a discrete dynamical system we are typically given the initial values of certain parameters and some information on how the parameters evolve in time. Usually the initial values of the parameters are encoded in a vector, say, u_0 and we are to describe the evolution of the system by finding the so-called *step matrix* for the system. The role of the step matrix, let's call it A, is to relate any two consecutive values (or states) of the parameters in the following way

$$u_{k+1} = Au_k, \quad k = 0, 1, 2, \ldots \tag{1}$$

That is, once we find A, if we know the state of the system at the k-th moment, we use (1) to determine the state of the system at the next, $(k+1)$-th, moment. For instance, since we know u_0 (that's data we are given), by using the identity (1) (which we can call a *recursive formula*), we find u_1 as follows

$$u_1 = Au_0,$$

and once we have u_1, again by using (1), we get u_2 by doing

$$u_2 = Au_1 = AAu_0 = A^2u_0.$$

And we find u_3 by doing

$$u_3 = Au_2 = AA^2u_0 = A^3u_0.$$

And at this point we realize that given any moment k, we can write u_k (the state of the system at the k-th moment) in terms of the initial state. Namely,

$$u_k = A^k u_0, \quad k = 0, 1, 2, \ldots \tag{2}$$

Steady state. Given a discrete dynamical system, there might be some values of the parameters involved that do not change from one moment to the next. That is, there might be a state vector u such that

$$u = Au. \tag{3}$$

A vector u satisfying (3) is called a *steady state* of the discrete dynamical system. We can get a sense of the existence of a steady state by looking at the values of u_k for large values of k. More accurately, we can just solve a linear system and determine whether the discrete dynamical system admits a steady state. How so? Well, we look at (3) and re-write it as

$$Iu = Au,$$

where I stands for the identity matrix (whose size is determined by the context of the problem). But, $Iu = Au$ is the same as $(A - I)u = 0$ (where the 0 is the zero column vector of the appropriate size). Therefore, in order to solve for u, we can do, for instance, using Matlab

```
>> ref([A-I, z])
```

(where z denote the zero vector mentioned above) and interpret. If there are free variables, they can also be determined by the context of the problem.

Let's see an example.

Example 1. Suppose that a rental car company has two locations: Downtown and Airport. In the beginning there are 120 cars in Downtown and 80 at the Airport. After repeated observations, it is noticed that, each week, 30 % of the cars picked up from Downtown are returned at the Airport and 10 % of the cars picked up from the Airport are dropped in the location Downtown. Determine the number of cars at each location by the 6th week.

In order to address this problem, or any other Math problem, we first introduce variables. Let d_k and a_k denote the number of cars in Downtown and at the Airport, respectively, in the k-th week. In particular, at the beginning, we have $d_0 = 120$ and $a_0 = 80$.

The next step, as in any other Math problem, is to relate the variables through equations deduced from the statement of the problem. For instance, we read that, in a given week the number of cars at each location is determined by those numbers in the preceding week. More precisely, looking at the Downtown location we see that, from one week to the next, 70 % of the cars picked up from Downtown will be returned to Downtown (the other 30 % will be at the Airport) and 10 % of those picked up from the Airport will be dropped in Downtown as well. We can put this information in terms of our variables as follows

$$d_{k+1} = 0.7d_k + 0.1a_k$$

Reasoning along the same lines, but now with the Airport location, we get

$$a_{k+1} = 0.3d_k + 0.9a_k$$

The next step is to write our equations in matrix form. Let's put all our parameters (d_k and a_k) in a vector by defining

$$u_k = \begin{bmatrix} d_k \\ a_k \end{bmatrix}$$

The vector u_k represents the state of the parameters at the k-the week. Notice that we have

$$u_0 = \begin{bmatrix} 120 \\ 80 \end{bmatrix} \quad \text{and} \quad u_{k+1} = \begin{bmatrix} d_{k+1} \\ a_{k+1} \end{bmatrix}.$$

By the definition of matrix multiplication, we see that our equations above can be recast as

$$u_{k+1} = \begin{bmatrix} d_{k+1} \\ a_{k+1} \end{bmatrix} = \begin{bmatrix} 0.7 & 0.1 \\ 0.3 & 0.9 \end{bmatrix} \begin{bmatrix} d_k \\ a_k \end{bmatrix} = Au_k. \tag{4}$$

That is, $u_{k+1} = Au_k$, where

$$A = \begin{bmatrix} 0.7 & 0.1 \\ 0.3 & 0.9 \end{bmatrix}$$

is the step matrix of this discrete dynamical system. Since the formula (2) says that

$$u_k = A^k u_0$$

we find the state of the system at the 6th week by doing, for instance, in Matlab,

```
>> A =[0.7 0.1; 0.3 0.9]

A =

    0.7000    0.1000
    0.3000    0.9000

>> u0=[120 80]'

u0 =

   120
    80

>> u6=A^6*u0

u6 =

   53.2659
  146.7341
```

Therefore, we conclude that at the end of the 6th week there will be 53.2659 cars in Downtown and 146.7341 cars at the Airport... Of course, these are just mathematical quantities that should be understood as a means to illustrate the method. A more realistic approach would be to round-off the state vectors at each intermediate step. That is, compute u_1, round it off (if necessary) to compute $u_2 = Au_1$ and so on.

Does the system have a steady state?

To get a sense of this we can look at the evolution of the system in the long run. For instance, by doing

```
>> u20=A^20*u0

u20 =

   50.0026
  149.9974

>> u50=A^50*u0

u50 =

   50.0000
  150.0000

>> u100=A^100*u0

u100 =

   50.0000
  150.0000

>> u200=A^200*u0

u200 =

   50.0000
  150.0000
```

Everything seems to indicate that this discrete dynamical system has the state vector

$$u = \begin{bmatrix} 50 \\ 150 \end{bmatrix}$$

as a steady state. We can easily verify this by checking that, indeed, $Au = u$, that is,

```
>> u=[50 150]'

u =

    50
   150

>> A*u
```

```
ans =

    50
   150
```

so that, in fact, $Au = u$. If we don't like the "guessing" part of this approach, we can rigorously deduce that the vector u above is a steady state by solving a system. Indeed, a steady state vector u must satisfy $Au = u$ (in other words, $(A - I)u = 0$) and the sum of its entries must equal 200 (the total number of cars). That is, we can do

```
I =

     1     0
     0     1

>> A-I

ans =

   -0.3000    0.1000
    0.3000   -0.1000

>> B=[A-I; 1 1]

B =

   -0.3000    0.1000
    0.3000   -0.1000
    1.0000    1.0000

>> d=[0 0 200]'

d =

     0
     0
   200

>> rref([B d])
```

```
ans =

     1     0    50
     0     1   150
     0     0     0
```

And we read that the solution is the vector u from above. This means that, in the long run, week after week there will be 50 cars in Downtown and 150 at the Airport. Of course, they need not be the same 50 or 150 cars each time.

By doing

```
>> u7=A^7*u0

u7 =

   51.9596
  148.0404

>> u8=A^8*u0

u8 =

   51.1757
  148.8243

>> u9=A^9*u0

u9 =

   50.7054
  149.2946

>> u10=A^10*u0

u10 =

   50.4233
  149.5767
```

we can predict that the system will reach its steady state after the 10th week.