DS102 - Discussion 10

Wednesday, 13th November, 2019

1. **Control of linear systems** In this question, we will look at the control of simple linear dynamical systems of the form:

$$x_{t+1} = Ax_t + Bu_t$$

Where $x \in \mathbb{R}^2$, $A \in \mathbb{R}^{2\times 2}$,$B \in \mathbb{R}^{2\times 1}$,$u \in \mathbb{R}$.

As we saw in class, linear models allow us to control more complicated systems like planes and quadcopters.

This question will rely on your knowledge of basic Linear Algebra quantities like eigenvectors and eigenvalues, as well as the concept of a norm.

(a) In this first part we will assume that there is no control input u, so that the dynamics are given by:

$$x_{t+1} = Ax_t$$

Derive an explicit formula of x_t given only A and x_0, the initial position.

Solution:

$$\begin{aligned} x_t &= Ax_{t-1} \\ &= A(Ax_{t-2}) \\ &\vdots \\ &= A^t x_0 \end{aligned}$$

(b) Suppose the matrix A is given by:

$$A = \begin{bmatrix} 1.2 & 1 \\ 0 & 0.2 \end{bmatrix},$$

For each initial condition given below, what is $\lim_{t\to\infty} ||x_t||$?

$$(i)\ \ x_0 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \qquad (ii)\ \ x_0 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \qquad (iii)\ \ x_0 = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

Solution: We note that the eigen-value, eigen-vector pairs, (λ, v) for A are given by:

$$({}_1, v_1) = \left(1.2, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right), ({}_2, v_2) = \left(0.2, \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right)$$

(i) Since x_0 is the eigenvector associated with eigenvalue 1.2, we have that:

$$\begin{aligned}\lim_{t\to\infty} ||x_t|| &= \lim_{t\to\infty} ||A^t x_0|| \\ &= \lim_{t\to\infty} 1.2^t ||x_0|| = \infty\end{aligned}$$

(ii) Since x_0 is the eigenvector associated with eigenvalue 0.2, we will have this go to infinity by the same argument:

$$\begin{aligned}\lim_{t\to\infty} ||x_t|| &= \lim_{t\to\infty} ||A^t x_0|| \\ &= \lim_{t\to\infty} 0.2^t ||x_0|| = 0\end{aligned}$$

(iii) Since x_0 is the sum of both eigenvectors under these dynamics, x_t will go to infinity by the same argument:

$$\begin{aligned}\lim_{t\to\infty} ||x_t|| &= \lim_{t\to\infty} ||A^t x_0|| \\ &= \lim_{t\to\infty} ||A^t(3v_1 - v_2)|| = \infty\end{aligned}$$

(c) Now suppose that the matrix A is given by:

$$A = \begin{bmatrix} 0.1 & 10 \\ 0 & 0.5 \end{bmatrix},$$

Show that for all initial conditions $x_0 \in \mathbb{R}^2$, $\lim_{t\to\infty} ||x_t|| = 0$. Such a system is called 'stable'.

Solution: We note that the matrix A has two eigenvalues both of which have magnitude less than 1. Let v_1 and v_2 be the eigenvectors associated with the eigenvalues and note that they must span $\mathbb{R}^2$.
For every initial condition $x_0 \in \mathbb{R}^2$, we can write it as:

$$x_0 = c_1 v_1 + c_2 v_2$$

For some scalars c_1 and c_2.

$$\begin{aligned}\lim_{t\to\infty} ||x_t|| &= \lim_{t\to\infty} ||A^t x_0|| \\ &= \lim_{t\to\infty} ||A^t(c_1 v_1 + c_2 v_2)|| \\ &= \lim_{t\to\infty} ||0.1^t c_1 v_1 + 0.2^t c_2 v_2)|| \\ &= 0\end{aligned}$$

(d) Now let us go back to the matrix A given by:

$$A = \begin{bmatrix} 1.2 & 1 \\ 0 & 0.2 \end{bmatrix},$$

Let us now assume that we have a control input to the system:

$$x_{t+1} = Ax_t + Bu_t$$

where B is either:

$$B = \begin{bmatrix} 6 \\ 0 \end{bmatrix} \qquad \text{or} \qquad B = \begin{bmatrix} -1.2 \\ 1.2 \end{bmatrix}$$

Which of these choices of B will allow you to stabilize the system? Given your choice can you come up with a stabilizing controller of the form:

$$u_t = -Kx_t,$$

where $K \in \mathbb{R}^{1\times 2}$ such that the system is stable?

Solution: If you choose $B = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$, and let $K = [K_1, 0]$ you get that the closed loop system is given by:

$$A - BK = \begin{bmatrix} 1.2 - 6.0K_1 & 1 \\ 0 & 0.2 \end{bmatrix}$$

If $K_1 = 1/5$, this gives:

$$A - BK = \begin{bmatrix} 0 & 1 \\ 0 & 0.2 \end{bmatrix}$$

Which is stable.

Note that the other choice would make it impossible to affect the unstable mode, and so it would be impossible to stabilize the system.

2. **Stable Matching**

In Lecture, we encountered the idea of a stable matching between two sides of a market. The classic example of matching is that of matching Medical Residents to Hospitals.

In this question, we will analyze properties of the Gale-Shapely Algorithm that was discussed in Lecture. To do so we first define the problem.

In the problem we consider, there are a set of n Hospitals, $h_1, ..., h_n$ and n Residents $r_1, ..., r_n$. Each Hospital has a ranked preference over all Residents and vice versa.

A matching M matches each Hospital i to a Resident j. A Matching is called *stable* if there is no hospital-resident pair that prefers each other to their current matches. We remark that there may be multiple stable matchings for the same set of Hospitals, Residents, and preferences.

We say a Resident r is *attainable* for a Hospital h if there exists a stable matching where they are matched.

In lecture we saw that the Gale-Shapley Algorithm computes a stable matching. Recall that it proceeds as follows:

1. Initialize all Hospitals and Residents as unmatched.
2. Each Hospital proposes to its most preferred Resident who has not rejected them yet.
3. Each Resident is temporarily matched to its most preferred Hospital out of those that propose to it and rejects the rest.
4. Steps (2) and (3) are repeated until there are no more rejections made and everyone is matched. At that point all temporary allocations become final.

Given this setup, we will analyze properties of the Gale-Shapley algorithm and stable matchings.

(a) Given the following set of three hospitals and residents and their preferences:

- $h_1 : r_1 > r_2 > r_3$.
- $h_2 : r_2 > r_3 > r_1$.
- $h_3 : r_1 > r_3 > r_2$.
- $r_1 : h_2 > h_3 > h_1$.
- $r_2 : h_2 > h_3 > h_1$.
- $r_3 : h_1 > h_2 > h_3$.

Is the matching $M = ((h_1, r_1), (h_2, r_2), (h_3, r_3))$ a stable matching?

Solution: For the matching M, we can see that resident r_1 and hospital h_3 both prefer each other to their partners since h_1 prefers r_1 to all other residents, and r_1 prefers h_3 to h_1.

(b) Show that the Gale-Shapley Algorithm matches each hospital to its most preferred attainable resident. (Hint: try assuming that the Gale-Shapley (GS) Algorithm does not match each hospital to its most preferred attainble resident and derive a contradiction using the definition of the algorithm).

Solution: To prove this, we will assume that it is not true and derive a contradiction. This proof technique is called proof by contradiction and can be very useful.

Suppose the GS Algorithm does not match h to its most preferred attainable resident r. Consider the first time a hospital h is rejected by its most preferred attainable resident r by the GS algorithm. For that to happen, r must prefer at that time a hospital h' more than h. (**this is true from the definition of the algorithm**).

Since this is the first time that a hospital is rejected by its most preferred attainable resident, h' must prefer r at least as much as its most preferred attainable resident, since this match will get fixed when the algorithm terminates. (**this is true from the definition of the algorithm**).

However, we know that r is the most preferred attainable resident of hospital h. That means that there exists a stable matching M' where h is matched to r. In that matching, h' must be matched to another resident r'' with rank less than r since r is its most preferred attainable match. (**this is true from the definition of most preferred attainable matches**).

Since, we have shown that h' and r must prefer each other to their mathches in M', it is not a stable matching. Therefore we have derived a contradiction, and out original premise must be wrong. Therefore, the stable matching M resulting from the GS Algorithm must be hospital-optimal.