36-720 Lecture on DYNAMICS ON NETWORKS

Scribed by: Xiaoying Tu

Lecture conducted on Oct 17, 2016

Contents

1 Reminders about linear dynamical systems

Given vector $\vec{x} \in \mathbb{R}^n$, linear dynamic has $\frac{d\vec{x}}{dt} = \mathbf{B}\vec{x}$ for some $n \times n$ matrix $\mathbf{B}$. In discrete time, this can also be written as:

$$\vec{x}(t+1) - \vec{x}(t) = \mathbf{B}\vec{x}(t)$$

or

$$\vec{x}(t+1) = (\mathbf{B}+\mathbf{I})\vec{x}(t) = \mathbf{C}\vec{x}(t)$$

As a nice matrix, $\mathbf{C}$ has at most n distinct eigenvalues with orthogonal eigenvectors. That is, $\mathbf{C}$ gives a system of n orthogonal eigenvectors, or a basis for $\mathbb{R}^n$. Let's call these eigenvectors $\vec{v_i}$ and the corresponding eigenvalues λ_i where $i \in 1:n$, then

$$\vec{x}(0) = \sum_{i=1}^{n} \vec{v_i}(\vec{x}(0) \cdot \vec{v_i})$$

and

$$\vec{x}(t) = \mathbf{C}^t \vec{x}(0)$$

This is called the evolution operator. Applying this gives us:

$$\vec{x}(1) = \mathbf{C}\vec{x}(0) = \mathbf{C}\sum_{i=1}^{n} \vec{v_i}(\vec{x}(0) \cdot \vec{v_i}) = \sum_{i=1}^{n}(\vec{x}(0) \cdot \vec{v_i})\mathbf{C}\vec{v_i} = \sum_{i=1}^{n}(\vec{x}(0) \cdot \vec{v_i})\lambda_i \vec{v_i}$$

And after evolving for t times:

$$\vec{x}(t) = \sum_{i=1}^{n}(\vec{x}(0) \cdot \vec{v_i})\lambda_i^t \vec{v_i}$$

The eigenvectors where $|\lambda_i| < 1$ go to 0 exponentially fast. The eigenvectors where $|\lambda_i| > 1$ go to ∞ exponentially fast.

2 Dynamics on Network I: Averaging or Diffusion

Think of some field x with a value x_i at each node i. The rate of change in x_i can be expressed as:

$$\frac{dx_i}{dt} = r \sum_{j \neq i} A_{ij}(x_j - x_i) \qquad (r > 0)$$

The dynamics behind this is the following. Each node looks at each of its neighbors. If the neighbor has more "x-stuff", we transfer some of the "x-stuff" to this node from the neighbor. Otherwise, if this node has more "x-stuff", it will transfer some to the neighbor. Overall, the "x-stuff" flows from high concentration to low, and the flow is along the graph. Eventually we even out the inequality.

2.1 The graph Laplacian

This process can be written equivalently in vector form as the following (for undirected graph):

$$\frac{d\vec{x}}{dt} = -r\mathbf{L}\vec{x}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{A}$ and $\mathbf{D} = diag(degrees)$. Similarly, this can be expressed in the form of discrete time.

Here, the matrix $\mathbf{L}$ is called the Laplacian of the graph. Recall that in multivariable calculus (continuous form):

$$\mathbf{L} = \frac{\partial^2 f(\cdot)}{\partial x^2} + \frac{\partial^2 f(\cdot)}{\partial y^2} + \frac{\partial^2 f(\cdot)}{\partial z^2}$$

The graph Laplacian on a "geometric" graph (gridded manifold) approaches the continuous $\mathbf{L}$ above.

2.2 Properties of the Laplacian

The Laplacian encodes lots of facts about the graph:

- $\mathbf{L}$ is symmetric (for undirected graph), which means all eigenvalues are real.
- One of these eigenvalues has to be 0 with eigenvector $\vec{1}$ regardless of the specific graph, since $\mathbf{L}\vec{1} = 0$ and $D_{ii} = \sum_j A_{ij}$
- All other eigenvalues are positive, i.e. all are going to be increasing.
- 0 can be a degenerate eigenvalue (i.e. having multiple eigenvectors). If w is a connected component, then $\vec{1}\mathbf{I_w}$ is also an eigenvector with zero-eigenvalue. This leads to a distinct 0 eigenvector for each connected component.
- Many, many other graph properties relate to the spectrum of $\mathbf{L}$. See the book Chung (1997) for more information.

2.3 Dividing the graph with the Laplacian

Assume (for simplicity) there is only one connected component in the graph. In this case, $\vec{1}$ is an eigenvector of $\mathbf{L}$ with eigenvalue 0. The next largest eigenvalue $\lambda_2 < 0$, and its eigenvector $\vec{v_2}$ must be orthogonal to $\vec{1}$. So $\vec{v_2} \cdot \vec{1} = \sum_{i=1}^{n} v_{2i} = 0$, which means $\vec{v_2}$ has both positive and negative entries. $\vec{v_2}$ divides the graph into two parts depending on the signs of the v_{2i} entries. This division is the one with the least flow between two parts. If we apply the evolution operator $\vec{x}(t+1) - \vec{x}(t) = -r\mathbf{L}\vec{x}(t)$ repeatedly, eventually we will have $\vec{x}(t) \to \alpha\vec{1}$ as the dominant eigenvector, while the $\vec{v_2}$ component of $\vec{x}(0)$ is the non-uniformity which survives the longest. Intuitively, running the averaging process will eventually put the graph into total homogeneity; during the process, it will show $\vec{v_2}$ as this is the longest surviving non-uniformity.

2.4 Smoothing with the Laplacian

Think of a problem of smoothing values on a graph: given observations $x_1, \ldots, x_n$on a graph, smooth them by shrinking each x_i towards its neighbors (e.g. for the purpose of de-noise). We can translate the problem to the following:

$$\min_{\tilde{x}} \|\tilde{x} - x\|_2^2 + \gamma \sum_{i,j} A_{ij} (\tilde{x_i} - \tilde{x_j})^2$$

where the first part is the MSE to minimize, and the second part is a function that penalizes rapid change. This objective function is equivalent to:

$$||\tilde{x} - x||_2^2 - 2\gamma \tilde{x}^T \mathbf{L} \tilde{x}$$

Solving the minimization problem gives us:

$$\hat{x} = (\mathbf{I} + 2\gamma \mathbf{L})^{-1} x$$

2.5 Laplacian smoothing is conservative

Observation:

$$0 = \sum_i \frac{dx_i}{dt} = \sum_i r \sum_j A_{ij} (x_j - x_i) = r \sum_i \sum_j A_{ij} (x_j - x_i)$$

In the expression above, each term appears twice with opposite signs, and the total amount of "x-stuff", $\sum_i x_i$ remains constant. This is why the process is also termed "diffusion".

3 Dynamics on Network II: Random Walk on the Graph

Assume the walker is at one node at each time point. It takes a random neighbor of a current node and moves to it. This is a Markov Chain on the nodes with transition matrix AD^{-1}, i.e. it takes each row of A and divide it by the degree of the node (each neighbor gets the same probability of being moved to).

There is an equilibrium distribution $\vec{\rho} \in \mathbb{R}^n$ s.t.

$$AD^{-1} \vec{\rho} = \vec{\rho} \Rightarrow (I - AD^{-1}) \vec{\rho} = 0$$

$$\begin{aligned}
&\Rightarrow (I - AD^{-1}) D D^{-1} \vec{\rho} = 0 \\
&\Rightarrow (D - A) D^{-1} \vec{\rho} = 0 \\
&\Rightarrow \mathbf{L} D^{-1} \vec{\rho} = 0
\end{aligned}$$

So $D^{-1}\vec{\rho}$ is a 0-eigenvector of $\mathbf{L}$. We know that, as the 0-eigenvector, $D^{-1}\vec{\rho} = \alpha \vec{1} \Rightarrow \vec{\rho} = \alpha D \vec{1}$. So if we let the random walker run forever, the probability of the walker landing on a node is proportional to its degree.

4 Dynamics on Network III: Infectious Processes:

Each node is in one of a discrete, finite set of states, including (at least) two states: "S" (susceptible) and "I" (infectious). The infection is passed on from I to S with some probability. Other possible states include "R" (recovered / removed), "E" (incubating), etc. The variants of this model are named by the state patterns it encodes: *S.I.R.*, *S.I.S.*, etc. But all variants start with everyone in S and a small number (say, 1) of individuals being infectious I.

4.1 Classic epidemic model and the logistic curve

If a population is "well-mixed", as S meets I,

$$\frac{dS}{dt} = -rSI$$

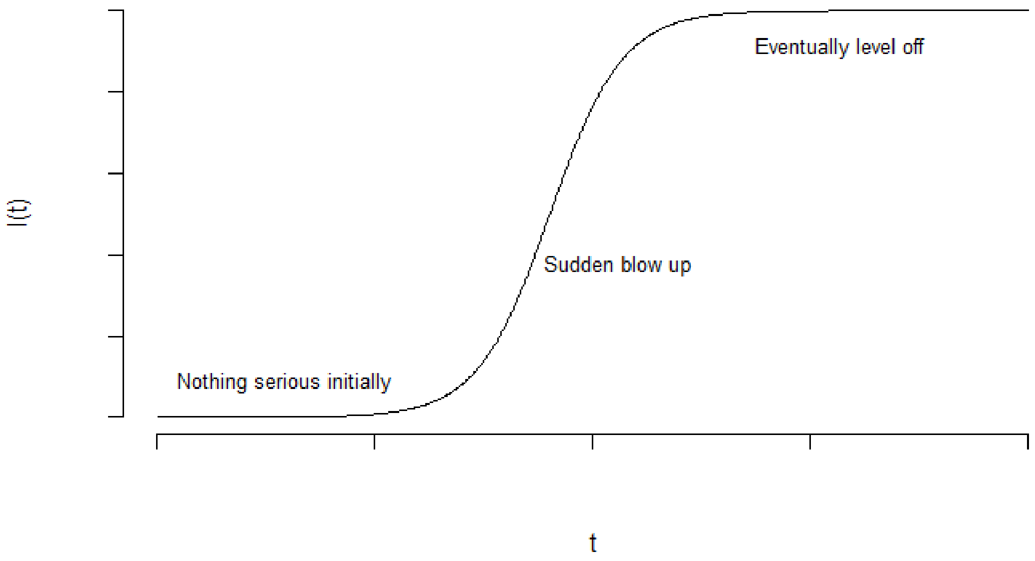
or

$$\frac{dI}{dt} = rI(N - I)$$

where N is the total population size. This indicates a pattern which shows exponential growth initially, but eventually levels off. The solution to the above ordinary differential equation is:

$$I(t) = \frac{Ne^{rt}}{N - I_0 + I_0 e^{rt}}$$

where $I_0 = I(0)$.



4.2 Branching process

A slightly more realistic version of the model is that we don't just look at the total numbers of S and I, but the status of each individual. Each node in the graph is, at any time, in one of the disease states. At each time, node i, if it is an I, has a certain probability of transmitting the disease to its S neighbors (if exist). Similarly, at each time, node j, if it is an S, has a certain probability of aquiring the disease from its I neighbors (if exist).

But how many individuals are infected at each time t? Assume t is small enough so that no one has yet recovered, which is equivalent to looking at how many individuals have ever been infected. We have

$$I(t) = Z(0) + Z(1) + \cdots + Z(t)$$

where $Z(t)$ is the number of newly infected at time t. Given that node i is S and its neighbor j is I, we define the average probability of transmission ("transmissibility") as τ.

How big is $Z(1)$? Each of the $Z(0)$ individuals infects $Binom(degree, \tau)$ of its neighbors. If we assume negligible overlaps, then $Z(1)$ is just the sum of all these $Z(0)$ binomials.

Recursively, $Z(t)$ is a sum of $Z(t-1)$ binomials, each of which being $Binom(degree - 1, \tau)$ (note that we need to subtract the degree by 1 to exclude the individual from whom it got the disease and who is already an I). We further define $u(t, i)$ to be the number of infections caused by individual i in t^{th} generation. Then

$$Z(t) = \sum_{i=1}^{Z(t-1)} u(t,i) = \sum_{i=1}^{Z(t-1)} Binom(degree_i - 1, \tau)$$

This is yet another branching process. This is equivalent to something called edge or bond percolation: take the original graph and flip a τ-coin for each edge; if it turns out heads (success), keep the edge; else if it turns out tails (failure), delete the edge. If $\tau \times$(expected number of new edges reached by following a random edge)> 1, then we will have a giant connected component which leads to epidemic. Otherwise, if $\tau \times$(expected number of new edges reached by following a random edge)< 1, the epidemic will die out.

References

Chung, Fan R. K. (1997). *Spectral Graph Theory*. Providence, Rhode Island: American Mathematical Society. URL `http://www.math.ucsd.edu/~fan/research/revised.html`.