Network Science		
Lecture 5: July 30, 2009	Scribe: Feng (Andrew) Ge	Lecturer: Henning Mortveit

Network Dynamical Process: SDS I

5.1 Introduction to Graph Dynamical Systems

Sequential Dynamic System (SDS) is now under a bigger umbrella called *Graph Dynamical System* (GDS), it finds applications in complex Systems such as:

- Epidemics on/over social contact graphs
- Vehicles in traffice networks (TRANSIMs)
- Biological cells in bio-networks
- Fighter units in war/surveillance settings

To study the above systems using GDS, we need to abstract common features among them, including:

- A collection of entities/agents
- Interactions among entities/agents
- Rules that specify outcomes of interactions
- An update scheme/mechanism specifying the manner of interactions

Accordingly, the mathematical abstraction of the above processes in GDS can be

- A graph where vertices represent entities and edges represent interaction possibilities
- A state for each vertex
- A rule/function for each vertex specifying how to transition from time t to time t+1 (locally)
- An update scheme scheduling the application of rules (locally)

Reflecting the complexities in the above applications, the GDS representation may have

- A large number of entities

- Stochastic rules which are hard to simulate real situations
- Update schemes: different schemes lead to different results, hard to explain.
- Co-evolving systems. For example, in an emergent situation of epidemics, there are three evolving systems: cellphone networks, vehicle traffic networks, and disease spreading networks.

We need to consider the process and relationship among three different things in science/engineering research: *real system*, *model*, and *model implementation*. One rule of thumb is that the model should reflect the real system well, at least for the aspects of your research focus. Usually this needs validatation. The model is usually mathematical equations. Further, model implementation may introduce a whole set of different problems. For instance, the update scheme plays an important role in GDS because different update scheme might derive to different results. This is the most uncertain step in SDS and GDS. In general, there are two types of update schemes: asynchronous update and synchronous update. The later one is more welcomed.

5.2 GDS Implementations

Let Y be a finit graph with the vertexset $V[Y] = 1, 2, ..., n$ and the edge set $E[Y]$. Then for each vertex i, we let $X_i \in K$ denote its states, where K is typical a finite set. We write $d(i)$ for the degree of vertex i, let $f_i : K^{d(i)+1} \to K$ be the vertex function for vertex i. This function reads the states of all i's neighbors and itself, then compute a state of i. We write $x = (x_1, x_2, ..., x_n)$ for the system state configuration, where x_i is the project of x to the i'th neighborhood.

Locally in isolation, we have
$x_i(t) \to f_i(x_i(t)) = x_i(t+1)$

The ith local map is
$F_i : K^n \to K^n$, where $F_i(x_1, ..., x_n) = (x_1, ..., x_{i-1}, f_i(x_(i)), ..., x_n)$

5.2.1 Generalized Cellular Automata (GCA)

Both GDS and SDS have their root in Generalized Cellular Automata (GCA) where update schemes apply all vertex functions simultaneiously and synchronously or in parallel.

The GCA map is given by $F : K^n \to K^n$, where $F(x_1, ..., x_n) = (f_1(x_1), ..., f_n(x_n))$

5.2.2 Sequential Dynamic Systems (SDS)

SDS is unique in that it follows a sequential update scheme. Let a permuation sequence $\pi = \pi_1, \pi_1, ..., \pi_n$, specify the order in which the vertex functions are applied. The SDS map is $F = F_{\pi_n} \cdot F_{\pi_{n-1}} ... \cdot F_{\pi_1}$.

One example is shown here. The graph is shown in Figure 5.1

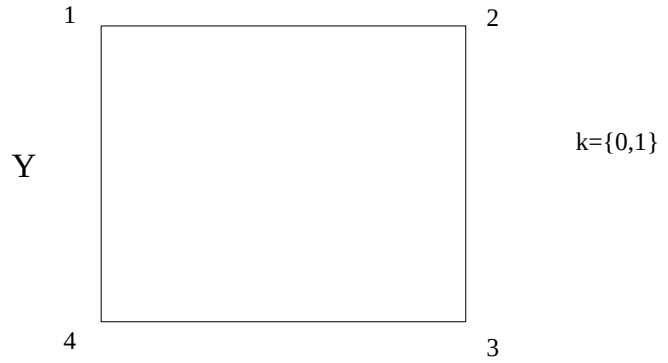

Figure 5.1: Example 1.

We define a *nor* function as $nor : K^3 \to K$

$$nor(x, y, z) = (1 + x)(1 + y)(1 + z) = \begin{cases} 1 & \text{if } x = y = z = 0 \\ 0 & \text{otherwise} \end{cases}$$

If we use a synchronous scheme in the udpate, we will have
$(0, 0, 0, 0) \to (1, 1, 1, 1)$.
However, if we use an asynchronous update scheme, for example, $\pi = (1, 2, 3, 4)$, we will have
$(0, 0, 0, 0) \to (1, 0, 0, 0) \to (1, 0, 1, 0) \to (1, 0, 1, 0)$.

The phase space contains all system state transition. Formally $\Gamma(F)$, with a vertex set $V[\Gamma(F)] = K^n$ and $e[\Gamma(F)] = (x, F(x)|x \in K^n)$. For all states, the transition diagram give the whole system configuration variations.

5.2.3 GCA and SDS Generalizations

If we introduce more dynamics into the models, we can have

- Time varying graph
- Stochastic aspect/elements
- More general udpate mechanisms