Bounded Confidence Model

Week 2-3

Paper 1

Bounded confidence model of opinion dynamics. Mostly describes the math behind the model.

- Describes and defines partial consensus. Gives certain condition which will almost surely result in partial consensus.
- Describes a cluster

Definition 3.6. We say that two peers i and j are *connected* at time k if their values x and y satisfy $|y - x| \leq \Delta$. We say that $F \subset \{1, 2, \dots, N\}$ is a *cluster* at time k if it is a maximal connected component.

Paper 1

Some interesting results:

- **Remark 4.1.** Each of the $\frac{N(N-1)}{2}$ unordered pairs of peers is thus chosen at rate $\frac{2}{N-1} = N/\frac{N(N-1)}{2}$, and then both peers undergo a *simultaneous* jump in their values if these are close enough. Each peer is thus affected at rate $2 = (N-1)\frac{2}{N-1}$.

- **Proposition 3.7.** *Let $\mathcal{C}^N(k) = \{C_1, \dots, C_\ell\}$ be the set of clusters at time k. Then either $\mathcal{C}^N(k+1) = \mathcal{C}^N(k)$ or $\mathcal{C}^N(k+1) = \left(\mathcal{C}^N(k) \setminus C_{\ell_1}\right) \cup \mathcal{C}'$ where $\mathcal{C}'$ is a partition of C_{ℓ_1}, for some $\ell_1 \in \{1, \dots, \ell\}$.*

  The number of clusters is thus non decreasing, and since it is bounded by $\lceil \frac{1}{\Delta} \rceil$ it must be constant after some time, yielding the following:

  Corollary 3.8. *There exists a random time K^N, a.s. finite, such that*

  $$\mathcal{C}^N(k) = \mathcal{C}^N(K^N) \text{ for } k \geq K^N .$$

Problems faced while building model based on this paper

- How do you update each node at each time step. The paper says it should be done at random. If random what should be the total time T for which we have to simulate.
- Topology and other specifications of the model are very convoluted. On the other hand, the mathematical proofs provided in this paper are useful.

I found a similar model, where the nodes are updated consistently at each time step which is easier to simulate.
NDlib - Network diffusion Library.

Other papers

1. [Modelling Group Opinion Shift to Extreme : the Smooth Bounded Confidence Model]()
   a. Discusses how polarization tend to shift to either polarization (2 clusters) or consensus (1 cluster) at the end of simulation.
   b. The method of update is

We first define :

$$g_u(x - x') = \exp\left(-\left(\frac{x - x'}{u}\right)^2\right) \tag{1}$$

This function is used to compute the influence of an encountered opinion $x'(t)$ on the individual's opinion $x(t)$ and uncertainty $u(t)$:

$$x(t+1) = \frac{x(t) + x'(t).g_u\left(x(t) - x'(t)\right)}{1 + g_u\left(x(t) - x'(t)\right)} \tag{2}$$

$$u(t+1) = \frac{u(t) + u'(t).g_u\left(x(t) - x'(t)\right)}{1 + g_u\left(x(t) - x'(t)\right)} \tag{3}$$

In this model, the function of influence is smooth when x' and u' vary. Moreover, small uncertainty opinions are more influential than high uncertainty opinions. This model shows the characteristics we were looking for by proposing the RA model, with a more direct mathematical expression.

NDlib

- NDlib is a Python language software package for the describing, simulate, and study diffusion processes on complex networks.
- Allows for custom model definition with a lot of freedom.
- At a higher level of abstraction a diffusion process can be synthesized into two components:
  - Available Statuses
  - Transition Rules that connect them
- Diffusion Models for opinion Dynamics. This pages explains most of what we need so far.

Weighted Hegselmann-Krause model

This model is a variation of the well-known Hegselmann-Krause (HK). During each interaction a random agenti is selected and the set Γ_ϵ of its neighbors whose opinions differ at most ϵ ($d_{i,j} = |x_i(t) - x_j(t)| \leq \epsilon$) is identified. Moreover, to account for the heterogeneity of interaction frequency among agent pairs, WHK leverages edge weights, thus capturing the effect of different social bonds' strength/trust as it happens in reality. To such extent, each edge $(i, j) \in E$, carries a value $w_{i,j} \in [0, 1]$. The update rule then becomes:

$$x_i(t+1) = \begin{cases} x_i(t) + \frac{\sum_{j\in\Gamma_\epsilon} x_j(t) w_{ij}}{\#\Gamma_\epsilon}(1 - x_i(t)) & \backslash\text{quad if x\_i(t) }\backslash\text{geq 0} \\ x_i(t) + \frac{\sum_{j\in\Gamma_\epsilon} x_j(t) w_{ij}}{\#\Gamma_\epsilon}(1 + x_i(t)) & \text{if x\_i(t) < 0} \end{cases}$$

The idea behind the WHK formulation is that the opinion of agent i at time $t + 1$, will be given by the combined effect of his previous belief and the average opinion weighed by its, selected, ϵ-neighbor, where $w_{i,j}$ accounts for i's perceived influence/trust of j.

Bounded confidence model

At each time tick two random agents are selected. If their opinions are not too different, $|x_i - x_j| < \epsilon$, they interact in the following manner:

a. $x_i(t+1) = x_i(t) - \mu(x_i(t) - x_j(t))$
b. $x_j(t+1) = x_j(t) + \mu(x_i(t) - x_j(t))$

Advantages of using Weighted Hegselmann-Krause

- Time steps are easily denoted and the update functions are easier to simulate
- Very similar to Bounded confidence model discussed
- NDlib has the model already built. It is a good starting point to export NDlib
  - Better understanding of the library
  - Helps build custom opinion dynamics model using the same library in the future
- Can explore different network structures since its just the matter of initializing in the library

CODE