Exploring Opinion Dynamics Filtering Strategies on Social Network Data

Zachary Zhao

Department of Computer Science and Engineering, Washington University in St. Louis

Masters Thesis Defense, 2023

Summary

Summary

- Background + Motivation

Summary

- Background + Motivation
- Introducing the Opinion Dynamics Model

Summary

- Background + Motivation
- Introducing the Opinion Dynamics Model
- Data Collection

Summary

- Background + Motivation
- Introducing the Opinion Dynamics Model
- Data Collection
- Data/Model Simulation

Summary

- Background + Motivation
- Introducing the Opinion Dynamics Model
- Data Collection
- Data/Model Simulation
- Conclusions and Final Remarks

Introduction to Opinion Dynamics

Early records of Opinion Dynamics

- John Milton's *Areopagitica* (1644)
  - Truth would prevail "in a free and open encounter".
- Justice Oliver Wendell Holmes in *Abrams v. United States*
  - Introduced "marketplace of ideas"

Spread of opinions especially prevalent now because of social media

Opinion Dynamics Setup

Opinion Dynamics seeks to model these interactions of ideas and opinions through modelling and quantitative analysis.

Opinion Dynamics Setup

Opinion Dynamics seeks to model these interactions of ideas and opinions through modelling and quantitative analysis.

Problems modeled by undirected graphs with weighted vertices.

- Each vertex is an agent
- Two vertices share an edge if the two agents have some relationship and can interact (ex. Facebook friends)
- Each vertex has a weight that represents an opinion
- Weights change over time

Filtering

Some interactions produce “filter bubbles”.

- set of users, all with similar opinions, interact more among themselves
- inspired by rise in virtual interactions through social networks

Filtering

Some interactions produce “filter bubbles”.

- set of users, all with similar opinions, interact more among themselves
- inspired by rise in virtual interactions through social networks

We also need to consider “filtering strategies”.

- social networks filter info, prioritize opinions aligned with an individual’s own opinions
- strategies are just algorithms that we use to do this filtering

Opinion Dynamics Model: Introduction

Model from Springsteen, Yeoh, and Vorobeychik (SYV) inspired by Tsang and Larson.

Initial Assumptions and Notation:

Opinion Dynamics Model: Introduction

Model from Springsteen, Yeoh, and Vorobeychik (SYV) inspired by Tsang and Larson.

Initial Assumptions and Notation:

- Undirected Graph $G = \langle V, E \rangle$ with weighted vertices

Opinion Dynamics Model: Introduction

Model from Springsteen, Yeoh, and Vorobeychik (SYV) inspired by Tsang and Larson.

Initial Assumptions and Notation:

- Undirected Graph $G = \langle V, E \rangle$ with weighted vertices
- Model observes the propagation of opinions over discrete set of timestamps $t \in \{0, 1, ..., T\}$.

Opinion Dynamics Model: Introduction

Model from Springsteen, Yeoh, and Vorobeychik (SYV) inspired by Tsang and Larson.

Initial Assumptions and Notation:

- Undirected Graph $G = \langle V, E \rangle$ with weighted vertices
- Model observes the propagation of opinions over discrete set of timestamps $t \in \{0, 1, ..., T\}$.
- $V = \{v_1, v_2, ...v_n\}$ with opinion at time $t = \{x_1^t, x_2^t, ..., x_n^t\}$

Opinion Dynamics Model: Introduction

Model from Springsteen, Yeoh, and Vorobeychik (SYV) inspired by Tsang and Larson.

Initial Assumptions and Notation:

- Undirected Graph $G = \langle V, E \rangle$ with weighted vertices
- Model observes the propagation of opinions over discrete set of timestamps $t \in \{0, 1, ..., T\}$.
- $V = \{v_1, v_2, ...v_n\}$ with opinion at time $t = \{x_1^t, x_2^t, ..., x_n^t\}$
- $\forall i, t,\ x_i^t \in [0, 1]$

Opinion Dynamics Model: Introduction

Model from Springsteen, Yeoh, and Vorobeychik (SYV) inspired by Tsang and Larson.

Initial Assumptions and Notation:

- Undirected Graph $G = \langle V, E \rangle$ with weighted vertices
- Model observes the propagation of opinions over discrete set of timestamps $t \in \{0, 1, ..., T\}$.
- $V = \{v_1, v_2, ... v_n\}$ with opinion at time $t = \{x_1^t, x_2^t, ..., x_n^t\}$
- $\forall i, t,\ x_i^t \in [0, 1]$
- 0 and 1 represent extreme opinions of an issue, 0.5 represents neutral opinion

Opinion Dynamics Model: Introduction

Model from Springsteen, Yeoh, and Vorobeychik (SYV) inspired by Tsang and Larson.

Initial Assumptions and Notation:

- Undirected Graph $G = \langle V, E \rangle$ with weighted vertices
- Model observes the propagation of opinions over discrete set of timestamps $t \in \{0, 1, ..., T\}$.
- $V = \{v_1, v_2, ...v_n\}$ with opinion at time $t = \{x_1^t, x_2^t, ..., x_n^t\}$
- $\forall i, t,\ x_i^t \in [0, 1]$
- 0 and 1 represent extreme opinions of an issue, 0.5 represents neutral opinion
- Edges in E notated as (v_i, v_j).

Opinion Dynamics Model: Introduction

Model from Springsteen, Yeoh, and Vorobeychik (SYV) inspired by Tsang and Larson.

Initial Assumptions and Notation:

- Undirected Graph $G = \langle V, E \rangle$ with weighted vertices
- Model observes the propagation of opinions over discrete set of timestamps $t \in \{0, 1, ..., T\}$.
- $V = \{v_1, v_2, ...v_n\}$ with opinion at time $t = \{x_1^t, x_2^t, ..., x_n^t\}$
- $\forall i, t, \; x_i^t \in [0, 1]$
- 0 and 1 represent extreme opinions of an issue, 0.5 represents neutral opinion
- Edges in E notated as (v_i, v_j).
- At each time t, v_i can share it's opinion with any of its neighbors. Let N_i be the set of neighbors for v_i.

Opinion Dynamics Model: Change of An Agent's Opinion

At $t = 0$, we initialize all opinions x_i^0. At timestamps $t > 0$, these opinions will evaluate according to

$$x_i^t \leftarrow \frac{deg(v_i)x_i^{t-1} + \sum_{v_j \in N_i} w_{ij}^{t-1} x_j^{t-1}}{deg(v_i) + \sum_{v_j \in N_i} w_{ij}^{t-1}} \tag{1}$$

where w_{ij}^{t-1} is a *trust value* maintained by agent v_i on agent v_j.

Opinion Dynamics Model: Change of An Agent's Opinion

At $t = 0$, we initialize all opinions x_i^0. At timestamps $t > 0$, these opinions will evaluate according to

$$x_i^t \leftarrow \frac{deg(v_i)x_i^{t-1} + \sum_{v_j \in N_i} w_{ij}^{t-1} x_j^{t-1}}{deg(v_i) + \sum_{v_j \in N_i} w_{ij}^{t-1}} \tag{1}$$

where w_{ij}^{t-1} is a *trust value* maintained by agent v_i on agent v_j.

Trust values are also initialized at $t = 0$ and update over time. In Tsang and Larson's model, they update as follows:

$$w_{ij}^t \leftarrow \frac{w_{ij}^{t-1} + rT(x_i^t, x_j^t)}{1 + r} \tag{2}$$

where r is a learning rate of the network, and T is a *trust function*.

Trust Function + Improvements on Tsang and Larson

Tsang and Larson use the following *trust function*:

$$T(x_i^t, x_j^t) = e^{-\frac{(x_i^t - x_j^t)^2}{h}} \tag{3}$$

where h is an empathy parameter via the Gaussian kernel.

Trust Function + Improvements on Tsang and Larson

Tsang and Larson use the following *trust function*:

$$T(x_i^t, x_j^t) = e^{-\frac{(x_i^t - x_j^t)^2}{h}} \tag{3}$$

where h is an empathy parameter via the Gaussian kernel.

Observe: Tsang and Larson's T is positive. This means agents will always move closer after an interaction. However, it could be the other way!

Trust Function + Improvements on Tsang and Larson

Tsang and Larson use the following *trust function*:

$$T(x_i^t, x_j^t) = e^{-\frac{(x_i^t - x_j^t)^2}{h}} \tag{3}$$

where h is an empathy parameter via the Gaussian kernel.

Observe: Tsang and Larson's T is positive. This means agents will always move closer after an interaction. However, it could be the other way!

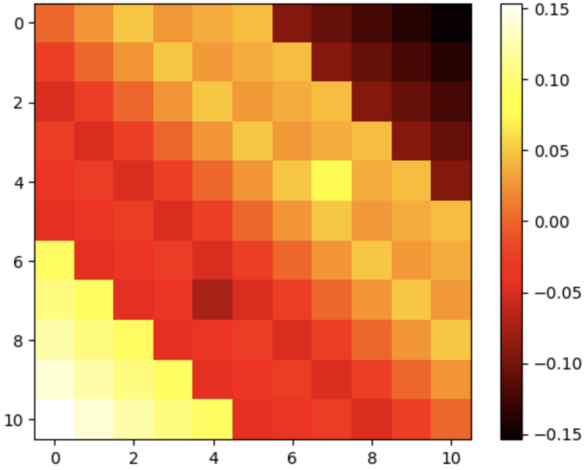
SYV: use *threshold parameters* d_1, d_2 such that $0 \leq d_1 \leq d_2 \leq 1$.

Three Zones

Given 2 agents v_i, v_j with opinions x_i^t, x_j^t:

- If $|x_i^t - x_j^t| < d_1$, w_{ij} increases, and v_i and v_j should move closer to each other. Call this an *assimilation* effect.
- If $d_1 \leq |x_i^t - x_j^t| \leq d_2$, w_{ij}^t remains the same and v_i and v_j will interact like how they did before time t. We call these interactions *neutral*.
- If $|x_i^t - x_j^t| > d_2$, w_{ij}^t decreases, and v_i and v_j move further away from each other. Call this a *boomerang* effect.

Visual of the Three Zones



Refinement of the Trust Function

We want to model opinions with a trust function that can now be negative, so we make the following modification to Tsang and Larson's T:

$$T(x_i^t, x_j^t) = \begin{cases} e^{-\frac{(|x_i^t - x_j^t| - d_1)^2}{(d_1/ln(2))^2}} & \text{if } |x_i^t - x_j^t| < d_1 \\ 1 - e^{-\frac{(|x_i^t - x_j^t| - d_2)^2}{((1-d_2)/ln(2))^2}} & \text{if } |x_i^t - x_j^t| > d_2 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

This T ranges from -1 to 1, so now have to alter the trust value update function as well:

$$w_{ij}^t \leftarrow \alpha w_{ij}^{t-1} + (1-\alpha) T(x_i^t, x_j^t) \tag{5}$$

for some learning rate $\alpha \in [0, 1]$.

Adding Filtering to the Model

Now we add filtering: At every timestamp, only have each agent v_i interact with $k < deg(v_i)$ of its neighbors based on a criterion.

Adding Filtering to the Model

Now we add filtering: At every timestamp, only have each agent v_i interact with $k < deg(v_i)$ of its neighbors based on a criterion.

- **Random Neighbors**: randomly select k neighbors of v_i.
- **Least Polar Neighbors**: select the k neighbors whose opinions are the closest to 0.5
- **Most Popular Neighbors**: select the k neighbors with the largest degrees
- **Most Similar Neighbors**: select the k neighbors whose opinions are closest to that of the agent

Adding Filtering to the Model

Now we add filtering: At every timestamp, only have each agent v_i interact with $k < deg(v_i)$ of its neighbors based on a criterion.

- **Random Neighbors**: randomly select k neighbors of v_i.
- **Least Polar Neighbors**: select the k neighbors whose opinions are the closest to 0.5
- **Most Popular Neighbors**: select the k neighbors with the largest degrees
- **Most Similar Neighbors**: select the k neighbors whose opinions are closest to that of the agent

If at time t no notable change of opinions occurs, then we say the network has *converged* and we stop iterating.

Data Collection

Two types of data were collected: **Facebook** and **Twitter** data.

Political party is opinion spectrum, with 0 being an extrema on the Democratic side and 1 being an extrema on the Republican side.

Data Collection

Two types of data were collected: **Facebook** and **Twitter** data.

Political party is opinion spectrum, with 0 being an extrema on the Democratic side and 1 being an extrema on the Republican side.

Facebook Network Data: "ego-Facebook" network from Stanford Network Analysis Project (SNAP).

Twitter Network Data: Tweepy and Twitter APIs calls, replicate Twitter network of US Senators

Facebook: Data Collection

SNAP Project ego-Network

- Anonymized network of social circles from Facebook
- 4039 Facebook users (labelled 0-4038), 88234 edges (labelled by pairs of vertices)
- Smaller networks present
- In small networks, one user, called an "ego", is connected to every other user in the small network

Facebook: Ego Network Properties

Each ego network also contains **features** for each user in the small network encoded as a binary string. We focus on the **gender** and **education** type features, shared between all ego networks.

```
0 birthday;anonymized feature 0
1 birthday;anonymized feature 1
2 birthday;anonymized feature 2
3 birthday;anonymized feature 3
4 birthday;anonymized feature 4
5 birthday;anonymized feature 5
6 birthday;anonymized feature 6
7 birthday;anonymized feature 7
8 education;classes;id;anonymized feature 8
9 education;classes;id;anonymized feature 9
10 education;classes;id;anonymized feature 10
11 education;classes;id;anonymized feature 11
12 education;classes;id;anonymized feature 12
```

```
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Facebook: Assigning Opinions with Probabilities

Wlog, let gender 1 be female, gender 2 be male, and education type 1-3 be some college, post-graduate degree, and a college degree respectively. No user had 0s for both gender types. If a user has 0s for all 3 education types, let the education type for that user be high school or less.

Facebook: Assigning Opinions with Probabilities

Wlog, let gender 1 be female, gender 2 be male, and education type 1-3 be some college, post-graduate degree, and a college degree respectively. No user had 0s for both gender types. If a user has 0s for all 3 education types, let the education type for that user be high school or less.

We make a probabilistic model that gives an opinion to a given node with a certain probability.

Facebook: Assigning Opinions with Probabilities

Wlog, let gender 1 be female, gender 2 be male, and education type 1-3 be some college, post-graduate degree, and a college degree respectively. No user had 0s for both gender types. If a user has 0s for all 3 education types, let the education type for that user be high school or less.

We make a probabilistic model that gives an opinion to a given node with a certain probability.

Pew Research Center Data: Proportions of the USA with various party affiliations (Conducted in 2014)

Affiliation	Women	Men	HS or Less	Some College	College Degree	Post-Grad
Democratic	39%	26%	32%	29%	35%	41%
Republican	25%	28%	30%	28%	24%	19%
Lean Dem	17%	18%	13%	18%	19%	22%
Lean Rep	12%	20%	17%	17%	15%	12%
Independent	7%	8%	8%	8%	7%	6%

Facebook: Assigning Opinions with Probabilities (cont.)

Given a user's education e and gender g and assuming that education and gender are independent, we can evaluate the user's party/affiliation p with the following equation:

$$p(p|e,g) = \frac{p(p|e)p(p|g)}{p(p)} \tag{6}$$

Facebook: Assigning Opinions with Probabilities (cont.)

Given a user's education e and gender g and assuming that education and gender are independent, we can evaluate the user's party/affiliation p with the following equation:

$$p(p|e,g) = \frac{p(p|e)p(p|g)}{p(p)} \tag{6}$$

$p(p)$ values are also extracted from the Pew Research Center, which are 33% for Democrats, 26% for Republicans, 17% for leaning Democrats, 16% for leaning Republicans, and 8% for Independents.

Facebook: Assigning Opinions with Probabilities (cont.)

Given a user's education e and gender g and assuming that education and gender are independent, we can evaluate the user's party/affiliation p with the following equation:

$$p(p|e,g) = \frac{p(p|e)p(p|g)}{p(p)} \tag{6}$$

$p(p)$ values are also extracted from the Pew Research Center, which are 33% for Democrats, 26% for Republicans, 17% for leaning Democrats, 16% for leaning Republicans, and 8% for Independents.

Then, a random number is generated between 0 and 1 and depending on what it is, an agent with given gender and education will start with that opinion for a certain trial. An opinion is 0 if the agent is Democratic, 1 if Republican, (0,0.5) if leaning Democratic, (0.5,1) if leaning Repulican, and 0.5 if Independent.

Twitter: Data Collection

Twitter network: US Senators on Twitter

Use Twitter API to gather Senators' Twitter IDs and connections.
Network has 100 nodes and 3765 edges.

Twitter: Data Collection

Twitter network: US Senators on Twitter

Use Twitter API to gather Senators' Twitter IDs and connections. Network has 100 nodes and 3765 edges.

Initial opinions are generated by Govtrack.us in their dataset titled "2022 Report Cards: All Senators / Ideology Score", which assigns every Senator a unique value from 0 to 1 with 0 being the most liberal and 1 being the most conservative.

Twitter: Data Collection (Govtrack Visual)

ALL SENATORS

most politically right

#1	1.00	Sen. Scott [R-FL]
#2	0.98	Sen. Braun [R-IN]
#3	0.94	Sen. Cruz [R-TX]
#4	0.93	Sen. Blackburn [R-TN]
#5	0.91	Sen. Cramer [R-ND]
#6	0.90	Sen. Lankford [R-OK]
#7	0.89	Sen. Lummis [R-WY]
#8	0.88	Sen. Marshall [R-KS]
#9	0.88	Sen. Inhofe [R-OK]
#10	0.88	Sen. Cotton [R-AR]
#11	0.87	Sen. Hyde-Smith [R-MS]
#12	0.86	Sen. Hawley [R-MO]
#13	0.86	Sen. Daines [R-MT]
#14	0.86	Sen. Hagerty [R-TN]
#15	0.83	Sen. Rubio [R-FL]
#16	0.83	Sen. Risch [R-ID]
#17	0.83	Sen. Ernst [R-IA]
#18	0.82	Sen. Crapo [R-ID]
#19	0.82	Sen. Tillis [R-NC]
#20	0.82	Sen. Barrasso [R-WY]

Twitter: Opinions Through Sentiment Analysis

In addition to the values provided by Govtrack.us, we also generate another set of opinions through Sentiment Analysis.

Twitter: Opinions Through Sentiment Analysis

In addition to the values provided by Govtrack.us, we also generate another set of opinions through Sentiment Analysis.

Tweepy API scrapes recent tweets of Senators and outputs a list. Then, we run each list through a series of cleansing methods.

```
testtweets = get_tweets("SenAlexPadilla")
testtweets
```

```
['I hear my Republican colleagues complain about inflation, rising prices, and workforce shortages. They know that o
u… https://t.co/KCcIsyQde6',
 'Voting is at the foundation of our democracy. When Americans are barred access from casting their ballots, it thre
a… https://t.co/1BACwhEkmO',
 'Happy Birthday Alejandro! Mommy and I can’t believe you’re turning 10 already. Feels like it was just yesterday th
a… https://t.co/mRtq3mks2U',
 '🗞NEWS: Over $35 million in federal funding will go towards reconnecting California communities torn apart by high
w… https://t.co/O5cZjfZ2Kp',
```

```
test_cleansed = cleanse(testtweets)
```

```
test_cleansed
```

```
['I hear my Republican colleagues complain about inflation rising prices and workforce shortages They know that ou',
 'Voting is at the foundation of our democracy When Americans are barred access from casting their ballots it threa',
 'Happy Birthday Alejandro Mommy and I cant believe youre turning 10 already Feels like it was just yesterday tha',
 'NEWS Over  million in federal funding will go towards reconnecting California communities torn apart by highw',
```

Twitter: Opinions Through Sentiment Analysis

After our list is outputted, we run each list through a series of cleansing methods, and apply the Vader sentiment analyzer from the NLTK package to analyze the sentiment of each tweet.

```
analyze(test_cleansed)
```

	neg	neu	pos	compound	sentiment
0	0.227	0.773	0.0	-0.4767	negative
1	0.0	1.0	0.0	0.0	neutral
2	0.0	0.721	0.279	0.7351	positive
3	0.118	0.882	0.0	-0.25	negative
4	0.0	0.761	0.239	0.5719	positive

NLTK **compound score** is weighted average over all positive and negative aspects of the tweet. Scores range from -1 to 1, so we apply scaling to get opinion between 0 and 1.

Running the Experiments: Preliminaries

Now that we have our network and can generate opinions, it's time to run the SYV model (all coded up, but I had to make many modifications to adjust the model to my networks). We use the following parameters and constraints:

Running the Experiments: Preliminaries

Now that we have our network and can generate opinions, it's time to run the SYV model (all coded up, but I had to make many modifications to adjust the model to my networks). We use the following parameters and constraints:

- Test all $d_1 \in \{0.3, 0.5, 0.7\}$, $d_2 \in \{0.5, 0.7, 0.9\}$ where $d_1 \leq d_2$. If we use another (d_1, d_2) combination, it will be mentioned for that particular experiment.

Running the Experiments: Preliminaries

Now that we have our network and can generate opinions, it's time to run the SYV model (all coded up, but I had to make many modifications to adjust the model to my networks). We use the following parameters and constraints:

- Test all $d_1 \in \{0.3, 0.5, 0.7\}$, $d_2 \in \{0.5, 0.7, 0.9\}$ where $d_1 \leq d_2$. If we use another (d_1, d_2) combination, it will be mentioned for that particular experiment.
- The trust values initialized to 1, and $\alpha = 0.001$.

Running the Experiments: Preliminaries

Now that we have our network and can generate opinions, it's time to run the SYV model (all coded up, but I had to make many modifications to adjust the model to my networks). We use the following parameters and constraints:

- Test all $d_1 \in \{0.3, 0.5, 0.7\}$, $d_2 \in \{0.5, 0.7, 0.9\}$ where $d_1 \leq d_2$. If we use another (d_1, d_2) combination, it will be mentioned for that particular experiment.
- The trust values initialized to 1, and $\alpha = 0.001$.
- Trial terminates when each agent's opinion changes no more than 10^{-3} or 500 iterations is reached.

Running the Experiments: Preliminaries

Now that we have our network and can generate opinions, it's time to run the SYV model (all coded up, but I had to make many modifications to adjust the model to my networks). We use the following parameters and constraints:

- Test all $d_1 \in \{0.3, 0.5, 0.7\}$, $d_2 \in \{0.5, 0.7, 0.9\}$ where $d_1 \leq d_2$. If we use another (d_1, d_2) combination, it will be mentioned for that particular experiment.
- The trust values initialized to 1, and $\alpha = 0.001$.
- Trial terminates when each agent's opinion changes no more than 10^{-3} or 500 iterations is reached.
- $k = 2$

Running the Experiments: Preliminaries

Now that we have our network and can generate opinions, it's time to run the SYV model (all coded up, but I had to make many modifications to adjust the model to my networks). We use the following parameters and constraints:

- Test all $d_1 \in \{0.3, 0.5, 0.7\}$, $d_2 \in \{0.5, 0.7, 0.9\}$ where $d_1 \leq d_2$. If we use another (d_1, d_2) combination, it will be mentioned for that particular experiment.
- The trust values initialized to 1, and $\alpha = 0.001$.
- Trial terminates when each agent's opinion changes no more than 10^{-3} or 500 iterations is reached.
- $k = 2$
- For each experiment involving randomness, we run 10 trials.

Running the Experiments: Preliminaries

Now that we have our network and can generate opinions, it's time to run the SYV model (all coded up, but I had to make many modifications to adjust the model to my networks). We use the following parameters and constraints:

- Test all $d_1 \in \{0.3, 0.5, 0.7\}$, $d_2 \in \{0.5, 0.7, 0.9\}$ where $d_1 \leq d_2$. If we use another (d_1, d_2) combination, it will be mentioned for that particular experiment.
- The trust values initialized to 1, and $\alpha = 0.001$.
- Trial terminates when each agent's opinion changes no more than 10^{-3} or 500 iterations is reached.
- $k = 2$
- For each experiment involving randomness, we run 10 trials.
- Filtering strategies used are Random Neighbors, Least Polar Neighbors, Most Similar Neighbors, and Most Popular Neighbors (Facebook only)

Results (Twitter, Sentiment Analysis)

Because of the tighter distribution of initial opinions, $d_1 = 0.1$ was also tried for this set of experiments. All experiments terminated quickly.

Results (Twitter, Sentiment Analysis)

Because of the tighter distribution of initial opinions, $d_1 = 0.1$ was also tried for this set of experiments. All experiments terminated quickly.

Random Neighbors: All opinions converged around 0.60 (the mean of the initial opinions). For $d_1 = 0.1$, we needed an average of 75.53 iterations. For $d_1 = 0.3$, we needed an average of 51.03 iterations. For $d_1 = 0.5$, we needed an average of 47.67 iterations. For $d_1 = 0.7$, we needed an average of 46.45 iterations.

Results (Twitter, Sentiment Analysis)

Because of the tighter distribution of initial opinions, $d_1 = 0.1$ was also tried for this set of experiments. All experiments terminated quickly.

Random Neighbors: All opinions converged around 0.60 (the mean of the initial opinions). For $d_1 = 0.1$, we needed an average of 75.53 iterations. For $d_1 = 0.3$, we needed an average of 51.03 iterations. For $d_1 = 0.5$, we needed an average of 47.67 iterations. For $d_1 = 0.7$, we needed an average of 46.45 iterations.

Least Polar Neighbors: All opinions converged around 0.5. $d_1 = 0.1$ converged after 92 iterations, $d_1 = 0.3$ after 59, $d_1 = 0.5$ after 55, and $d_1 = 0.7$ after 54.

Results (Twitter, Sentiment Analysis)

Because of the tighter distribution of initial opinions, $d_1 = 0.1$ was also tried for this set of experiments. All experiments terminated quickly.

Random Neighbors: All opinions converged around 0.60 (the mean of the initial opinions). For $d_1 = 0.1$, we needed an average of 75.53 iterations. For $d_1 = 0.3$, we needed an average of 51.03 iterations. For $d_1 = 0.5$, we needed an average of 47.67 iterations. For $d_1 = 0.7$, we needed an average of 46.45 iterations.

Least Polar Neighbors: All opinions converged around 0.5. $d_1 = 0.1$ converged after 92 iterations, $d_1 = 0.3$ after 59, $d_1 = 0.5$ after 55, and $d_1 = 0.7$ after 54.

Closest Neighbors: The network converged but most opinions barely moved. The mean opinions after termination was around 0.60. For $d_1 = 0.3$, we needed 44 iterations. For $d_1 = 0.5$, we needed 37 iterations. For $d_1 = 0.7$, we needed 36 iterations. For $d_1 = 0.1$, we only needed 5!

Results (Twitter, Govtrack)

All experiments terminated before 500 iterations.

Results (Twitter, Govtrack)

All experiments terminated before 500 iterations.

Random Neighbors: All opinions converged around some opinion with very little variance. For $d_1 = 0.3$, the opinions converged around 0.71. For $d_1 = 0.5$, it's around 0.67. For $d_1 = 0.7$, it's around 0.631. Iterations didn't really follow a pattern.

Results (Twitter, Govtrack)

All experiments terminated before 500 iterations.

Random Neighbors: All opinions converged around some opinion with very little variance. For $d_1 = 0.3$, the opinions converged around 0.71. For $d_1 = 0.5$, it's around 0.67. For $d_1 = 0.7$, it's around 0.631. Iterations didn't really follow a pattern.

Least Polar Neighbors: All opinions converged around 0.5. For $d_1 = 0.3$, we needed 219 iterations. For $d_1 = 0.5$, we needed 102 iterations. For $d_1 = 0.7$, we needed 84 iterations.

Results (Twitter, Govtrack)

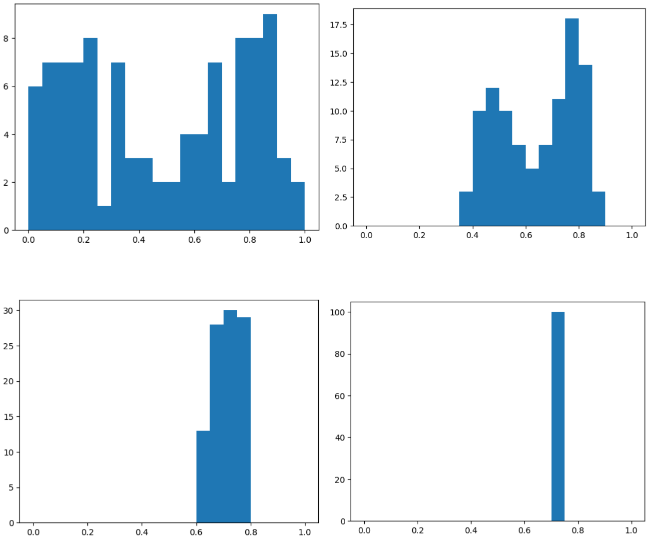
All experiments terminated before 500 iterations.

Random Neighbors: All opinions converged around some opinion with very little variance. For $d_1 = 0.3$, the opinions converged around 0.71. For $d_1 = 0.5$, it's around 0.67. For $d_1 = 0.7$, it's around 0.631. Iterations didn't really follow a pattern.

Least Polar Neighbors: All opinions converged around 0.5. For $d_1 = 0.3$, we needed 219 iterations. For $d_1 = 0.5$, we needed 102 iterations. For $d_1 = 0.7$, we needed 84 iterations.

Closest Neighbors: The network converged but most opinions barely moved. The mean opinions after termination was around 0.47. For $d_1 = 0.3$, we needed 38 trials. For $d_1 = 0.5$, we needed 120 trials. For $d_1 = 0.7$, we needed 39 trials.

Example of Opinion Distributions Visual (Random Neighbors)



Remarks (Twitter)

Overall observations and conjectures:

- If distribution of opinions is condensed enough, only d_1 matters

Remarks (Twitter)

Overall observations and conjectures:

- If distribution of opinions is condensed enough, only d_1 matters
- If distribution of opinions is really condensed, d_1 acts as a "learning rate"

Remarks (Twitter)

Overall observations and conjectures:

- If distribution of opinions is condensed enough, only d_1 matters
- If distribution of opinions is really condensed, d_1 acts as a "learning rate"
- If distribution of opinions is condensed enough, network more likely to terminate.

Remarks (Twitter)

Overall observations and conjectures:

- If distribution of opinions is condensed enough, only d_1 matters
- If distribution of opinions is really condensed, d_1 acts as a "learning rate"
- If distribution of opinions is condensed enough, network more likely to terminate.
- If network is dense, more likely for opinions to converge somewhere with Random and Least Polar

Remarks (Twitter)

Overall observations and conjectures:

- If distribution of opinions is condensed enough, only d_1 matters
- If distribution of opinions is really condensed, d_1 acts as a "learning rate"
- If distribution of opinions is condensed enough, network more likely to terminate.
- If network is dense, more likely for opinions to converge somewhere with Random and Least Polar
- Random Neighbors affected more by connections, not by mean opinion.

Remarks (Twitter)

Overall observations and conjectures:

- If distribution of opinions is condensed enough, only d_1 matters
- If distribution of opinions is really condensed, d_1 acts as a "learning rate"
- If distribution of opinions is condensed enough, network more likely to terminate.
- If network is dense, more likely for opinions to converge somewhere with Random and Least Polar
- Random Neighbors affected more by connections, not by mean opinion.
- Most Similar likely doesn't move much if enough connections are present, form "filter bubble"

Results (Facebook)

Note: Mean and spread of initial opinions are different for each trial.

Results (Facebook)

Note: Mean and spread of initial opinions are different for each trial.

Random Neighbors: All opinions converged around some opinion with very little variance. In general, the opinions would converge to the left if more 0s were initialized, and vice versa. Most trials took over 400 iterations, and some trials did not even converge before 500 iterations was hit. The behavior doesn't seem to be affected by d_1 and d_2 values.

Results (Facebook)

Note: Mean and spread of initial opinions are different for each trial.

Random Neighbors: All opinions converged around some opinion with very little variance. In general, the opinions would converge to the left if more 0s were initialized, and vice versa. Most trials took over 400 iterations, and some trials did not even converge before 500 iterations was hit. The behavior doesn't seem to be affected by d_1 and d_2 values.

Least Polar Neighbors: Most trials resulted in opinions converging around 0.5 with little variance. Most trials also terminated before 500 iterations, but the variances remained small but not too small. Otherwise, the opinions seemed to stay put. As d_1 increased, more of the first type of trial were found, and as d_2 increased for a fixed d_1, more of the second type were being found. An increase in d_1 lowered the number of average trials, with 305.90 for $d_1 = 0.3$, 226.78 for $d_1 = 0.5$ and 175.5 for $d_1 = 0.7$.

Results (Facebook) (cont.)

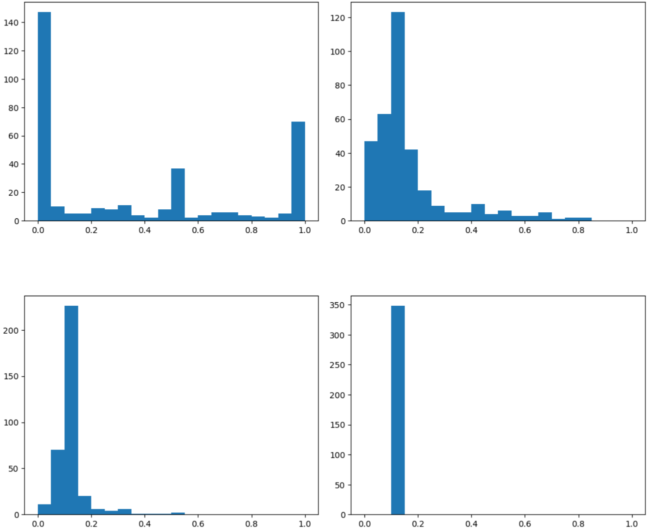
Closest Neighbors: Most opinions barely moved. A good portion of the trials converged after less than 100 iterations, but the rest all went up to 500. This strategy doesn't seem to be affected by d_1 and d_2.

Results (Facebook) (cont.)

Closest Neighbors: Most opinions barely moved. A good portion of the trials converged after less than 100 iterations, but the rest all went up to 500. This strategy doesn't seem to be affected by d_1 and d_2.

Most Popular Neighbors: Pretty much all opinions converged around some opinion with very little variance. Most trials converged around the ego node's initial opinion, however sometimes the opposite was the case, especially if the ego node was initialized to 0 or 1 or more opinions were initialized away from the ego node. The behavior of this strategy initially doesn't seem to be affected by d_1 and d_2 values but after observing what happens with both a larger d_1 and larger d_2, it appears that the opinions are more likely to converge to the ego node, which makes sense.

Example of Opinion Distributions Visual (Random Neighbors)



Remarks (Facebook)

Overall observations and conjectures:

Remarks (Facebook)

Overall observations and conjectures:

- Distribution of opinions matters the most if we fix a filtering strategy

Remarks (Facebook)

Overall observations and conjectures:

- Distribution of opinions matters the most if we fix a filtering strategy
- If distribution follows some pattern, THEN d_1 and d_2 will matter

Remarks (Facebook)

Overall observations and conjectures:

- Distribution of opinions matters the most if we fix a filtering strategy
- If distribution follows some pattern, THEN d_1 and d_2 will matter
- As long as the graph is dense enough, Least Polar is powerful

Remarks (Facebook)

Overall observations and conjectures:

- Distribution of opinions matters the most if we fix a filtering strategy
- If distribution follows some pattern, THEN d_1 and d_2 will matter
- As long as the graph is dense enough, Least Polar is powerful
- Random Neighbors tends to cause convergence if a lot of interactions are possible

Remarks (Facebook)

Overall observations and conjectures:

- Distribution of opinions matters the most if we fix a filtering strategy
- If distribution follows some pattern, THEN d_1 and d_2 will matter
- As long as the graph is dense enough, Least Polar is powerful
- Random Neighbors tends to cause convergence if a lot of interactions are possible
- If one or several nodes are the most popular neighbors for every node in our network, then they will have by far the most influence, but not all of it.

Remarks (Facebook)

Overall observations and conjectures:

- Distribution of opinions matters the most if we fix a filtering strategy
- If distribution follows some pattern, THEN d_1 and d_2 will matter
- As long as the graph is dense enough, Least Polar is powerful
- Random Neighbors tends to cause convergence if a lot of interactions are possible
- If one or several nodes are the most popular neighbors for every node in our network, then they will have by far the most influence, but not all of it.
- The larger the graph, the longer the graph takes to converge.

Conclusions and Final Remarks

Final Remarks

Conclusions and Final Remarks

Final Remarks

- For the filtering strategies that aren't Closet Neighbor and large enough assimilation zones, the network converges.

Conclusions and Final Remarks

Final Remarks

- For the filtering strategies that aren't Closet Neighbor and large enough assimilation zones, the network converges.
- In general, larger boomerang or neutral zones tends to only slow down the inevitable if enough opinions interact to assimilate.

Conclusions and Final Remarks

Final Remarks

- For the filtering strategies that aren't Closet Neighbor and large enough assimilation zones, the network converges.
- In general, larger boomerang or neutral zones tends to only slow down the inevitable if enough opinions interact to assimilate.
- In general, filtering strategy matters the most out of any factor/parameter

Conclusions and Final Remarks

Final Remarks

- For the filtering strategies that aren't Closet Neighbor and large enough assimilation zones, the network converges.
- In general, larger boomerang or neutral zones tends to only slow down the inevitable if enough opinions interact to assimilate.
- In general, filtering strategy matters the most out of any factor/parameter
- If the distribution is condensed enough, changing d_1 and d_2 slightly changes it

Conclusions and Final Remarks

Final Remarks

- For the filtering strategies that aren't Closet Neighbor and large enough assimilation zones, the network converges.
- In general, larger boomerang or neutral zones tends to only slow down the inevitable if enough opinions interact to assimilate.
- In general, filtering strategy matters the most out of any factor/parameter
- If the distribution is condensed enough, changing d_1 and d_2 slightly changes it
- Overall, just so many parameters to consider, so it's hard to predict the behavior of a certain initialized network. But we can try to guess the trend, especially if all the parameters follow a pattern.

Conclusions and Final Remarks

Final Remarks

- For the filtering strategies that aren't Closet Neighbor and large enough assimilation zones, the network converges.
- In general, larger boomerang or neutral zones tends to only slow down the inevitable if enough opinions interact to assimilate.
- In general, filtering strategy matters the most out of any factor/parameter
- If the distribution is condensed enough, changing d_1 and d_2 slightly changes it
- Overall, just so many parameters to consider, so it's hard to predict the behavior of a certain initialized network. But we can try to guess the trend, especially if all the parameters follow a pattern.
- Overall, assimilation seems to dominate especially in denser graphs.

Reflections + Further Work

Overall, the project was really fun and got me to dip my feet in a topic I wasn't entirely familiar with, Opinion Dynamics, but also allowed me to connect it to a topic I'm more used to, which is Data Science. In addition, I loved applying and integrating my prior knowledge of modelling and Graph Theory to something that can be applied to the real world.

Further work is being done into exploring other networks, such as the other ego networks, the entire large Facebook network, and a Twitter Network modelling the House of Representatives. We also want to attempt to prove some convergence properties mathematically.

Acknowledgements

I would like to thank the CSE Department for nurturing and educating me in computer science for the last five years, the Master's program for the amazing opportunity to do interesting research, and Professor William Yeoh and Jean Springsteen for being amazing mentors. In addition, I want to thank my peers and professors in the CSE department for keeping me on my toes and pushing me to be a better problem solver and scholar. Finally, I want to thank my parents for being extremely supportive in all my endeavors.

Thank You!