**Model Author:** Francesko

**Variants:** groupFormation, groupFormation\_identity

### groupFormation.nlogo documentation

#### Breeds

1. **Turtles**
2. **Centroids**
3. **Comms (undirected link)**
4. **l-distances (undirected link)**

#### Comms Attributes:

1. **op-weight**

#### L-distances Attributes:

1. **l-weight**

#### Turtle Attributes:

1. **Opinion-position :** the position of a point in the n-dimensional belief space
2. **P-speaking :** Probability of an agent speaking at a given point. Drawn as per p-speaking-drawn. Updated every step if p-speaking-drawn = “function”, otherwise set only once in setup in get-speaking.
3. **Speak? :** Whether agent speaks at a given time. Set in ‘speaking’
4. **Uncertainty :** Variable component of the HK variable epsilon. Set once in setup in ‘get-uncertainty’.
5. **Record:** An array of values containing whether the opinion array remained unchanged or not in the last n timesteps where n = ‘record-length’.
6. **Last-opinion:** Stores the just previous opinion ie (t-1). Updated every time step
7. **Pol-bias**
8. **Initial-opinion:**
9. **Satisfied?:** Whether the agent’s criteria for staying within current comms sub-network is satisfied. Computed every step. If unsatisfied, agent leaves community OR makes incremental changes to their network (param: network-change).
10. **Conformity:**
11. **group:** Index of current identity group
12. **Group-threshold:** *Francesko: “;; Individual sensitivity for group tightness/threshold.”*
13. **Identity-group:** set of agents constituting current identity group
14. **Opponents-ratio:** Current ratio within comms neighbourhood of agents falling beyond own HK boundary (opponents) to those within own HK boundary,
15. **Tolerance:** Maximum opponents-ratio from

#### Global Variables:

1. **p-speaking-drawn**

### Pseudocode - Base:

**Setup:**

Avoid Redundancies

Set Random Seed

Generate watts-strogatz network

Set Random Seed (Why again? Francesko: “To avoid some random artificialities due to small-world network generation, we have to set random seed again.”)

Ask Turtles to:

Initialise Opinion vector at random.

Initialise Initial-opinion

Set Last-opinion

Initialise Record

Initialise P-speaking

Set speak? Ie decide if to speak on the first tick

Initialise Pol-bias

Initialise Uncertainty

Initialise main-Record

Set file name core

Record current state of sim.

**Go:**

Avoid Redundancies

Ask turtles to:

If P-speaking must be dynamically updated:

Update P-speaking

Update speaking? Ie decide if to speak

Update Last-opinion

Implement opinion change mechanism as per ‘change-opinion-HK’

Update Record ie history of whether opinion change has occurred

Update main-Record as mean across turtles of the mean of their Records

### Pseudocode - Social Identity:

**Setup:**

Avoid Redundancies

Set Random Seed

Generate watts-strogatz network with comms as links

Set Random Seed (Why again? Francesko: “To avoid some random artificialities due to small-world network generation, we have to set random seed again.”)

Ask Turtles to:

Initialise Opinion vector at random.

Set Last-opinion

Initialise Record

Initialise P-speaking

Set speak? Ie decide if to speak on the first tick

Initialise Uncertainty

Initialise tolerance

Initialise conformity

Initialise group threshold

Initialise identity-group as empty turtle set

Initialise opponents-ratio as 0

Set ‘agents’ as set of turtles

Ask agents to create l-distances with other agents (creating a “full” network)

Ask l-distances to be hidden

Set comms to hidden

Initialise main-Record

Update-links-weights (?)

If use\_identity?:

Conditionally Set identity groups locally or globally

Else:

Set all agents as identity group of each agent

Update satisfaction of each agent

Set network-changes as 0 (?)

compute-polarization-repeatedly

Set file name core

Record current state of sim.

**Go:**

Avoid Redundancies

Prepare-everything-for-step

Ask turtles to:

Check satisfaction

If unsatisfied: change-of-network

*After this you use opponents ratio (alternately, dissatisfied\_updates\_opinion) OR satisfied? To determine whether agents opinion will change*

If P-speaking must be dynamically updated:

Update P-speaking

Update speaking? Ie decide if to speak

Update Last-opinion

Implement opinion change mechanism as per ‘change-opinion-HK’

Update Record ie history of whether opinion change has occurred

Connect Loners

recording-situation-and-computing-polarisation

### Misc. functions, params and variable info

**Functions:**

1. **Change-of-network (turtle):** Called when an agent is unsatisfied and needs to change their network. If param **network-change** is set to “link”, calls rewire-one-link else to “community”, calls leave-the-neighborhood-join-a-new-one.
2. **Rewire-one-link (turtle):** First kills one link to speaking agent in identity group - either at random, or to the most dissimilar such agent (param: cut-links-randomly?). Then select a speaking agent who isn’t linked already, either at random, or to the most similar such agent (param: create-links-randomly?).
3. **Leave-the-neighborhood-join-a-new-one (turtle):**

Leaves current neighbourhood, then forms a new neighbourhood of equivalent size. This can be done either randomly, or by choosing the nearest agents (param: create-links-randomly?)

Doubt: Why is nei-size being compared to speaking-others? Shouldn’t the nei-size always be less than or equal to number of non-self speaking agents?

1. **Set-group-identities:** 
   1. Select agents that have at least two neighbours within an opinion threshold (id\_treshold)

Doubt: Why not run the community detection on the entire network - why select well connected agents? Is the assumption here that communities are established only by connected agents?

* 1. Run Louvain community detection on the subgraph containing only these selected agents and their links.

Doubt:.Is the set-context function called on the correct subgraph here?

* 1. Compute centroids of each community.
  2. Assign ALL (not just well connected) agents to community with nearest centroid, and recompute centroids.
  3. Run k-means clustering on communities with k = number of communities so far (ie, iterate step d until centroids stop changing).

Doubt:.Why run both Louvian and k-means? Can’t we take only Louvian communities

* 1. Define “Identity-group” for each agent, ie an agentset of its own identity group. If this group is too small, let the entire set of individuals be its identity group instead.
  2. Remove centroids that have no agents associated.

1. **compute-centroid-positions : takes a list of agents as argument.**

Reset each centroid position to the mean opinion of subset of input agents with the appropriate centroid.

**Variables:**

1. **Network-changes (global):** Tracks number of changes in the network *in a single step*. Initialised to 0 in both setup and go, increments by 1 on every call of change-of-network. Used for checking sim termination conditions.
2. **network-change (UI parameter):** determines whether network change occurs via community or via links. Needs renaming to avoid confusion with network-changes

**openly-listen vs vaguely-speak:**

Distribution of ‘uncertainty’ is basically the distribution of the HK parameter epsilon. Epsilon is uncertainty scaled by some factor.

In openly listen mode, agents have a tolerance limit for whom they listen to based on opinion distance. In vaguely speak mode, agents have a tolerance limit for whom they speak to based on opinion distance.

QUESTION: Why is epsilon defined the way it is, ie uncertainty\*sqrt(4\*numberOfOpinionDimensions)

### Ideas for a speech-based dynamics model

Caution: this might complicate the model quite a bit.

But this strikes me as an interesting modification, motivated by two ideas:

(a) What could be called the “joe rogan phenomenon” where someone quite uninformed about a subject feels empowered to speak about it when the environment does not challenge it sufficiently. The interesting aspect of this phenomenon is that one’s own speech seems to strengthen one’s position on something. This is in contrast to the stereotypical presentation of radicalisation which involves diving deep about a subject and developing a mostly coherent ideology by reading books of a certain literary standard - it seems like the act of articulating a position, regardless of how naive or moderate or uninspiring it is, reinforces strength in said belief. Pre-internet it was harder to find peer groups where you can voice your first uninformed knee-jerk reaction to something without at least mild consequences. Something about

(b) Ashley mentioning how her own non-conservative views might have strengthened when exposed to hostile conversations with conservatives. The act of defending oneself might trigger “moving away” from the dominant viewpoint in the current conversation.

The issue here is that this requires a confidence dimension for each belief. Or at least some proxy for it.

Two ways to model this:

1. Beliefs/opinions are probability distributions rather than point values. Distributions change when opinions are heard, as well as expressed.
2. One stores a history of previously expressed (in speech) beliefs.

I am also afraid this is too complicated while being insufficiently justified. It lacks parsimony in some sense so far.