Genetic Programming for Understanding Cognitive Biases that Generate Polarization in Social Networks

First Author
Second Author

ABSTRACT

Recent studies have applied agent-based models to infer human-interpretable explanations of individual-scale behaviors that generate macro-scale patterns in complex social systems. Genetic programming has proven to be an ideal explainable AI tool for this purpose, where primitives may be expressed in an interpretable fashion and assembled into agent rules. Evolutionary model discovery is a tool that combines genetic programming and random forest feature importance analysis, to infer individual-scale, human-interpretable explanations from agent-based models. We deploy evolutionary model discovery to investigate the cognitive biases behind the emergence of ideological polarization within a population. An agent-based model is developed to simulate a social network, where agents are able to create or sever links with one another, and update an internal ideological stance based on their neighbors' stances. Agent rules govern these actions and constitute of cognitive biases. A set of 7 cognitive biases are included as genetic program primitives in the search for rules that generate hyper-polarization among the population of agents. We find that heterogeneity in cognitive biases is more likely to generate polarized social networks. Highly polarized social networks are likely to emerge when individuals with confirmation bias are exposed to those with either attentional bias or egocentric bias.

CCS CONCEPTS

• Computing methodologies → Genetic programming; **Agent / discrete models**; **• Applied computing → Sociology**.

KEYWORDS

Genetic Programming, Agent-based, Cognitive Bias, Polarization, Social Network

ACM Reference Format:
First Author and Second Author. 2022. Genetic Programming for Understanding Cognitive Biases that Generate Polarization in Social Networks.

GECCO 2022, July 9–13, 2022, Boston, MA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

In *Proceedings of GECCO 2022: The Genetic and Evolutionary Computation Conference (GECCO 2022).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Agent-based models are useful for simulating real-world complex phenomena due to their ability to model individual-scale interactions. In recent years, there have been several efforts to use agent-based models to infer the individual-scale behaviors that give rise to macro-scale phenomena of interest[1, 17, 33]. Of particular importance is human-interpretability of the inferred behaviors. Genetic programming has proved to be a ideal explainable AI tool enabling this goal due the ability of terminals to be expressed in human-interpretable units, which in turn is assembled into human-interpretable individual-scale behaviors. Evolutionary model discovery is a tool that combines genetic programming and random forest feature importance analysis, to infer individual-scale human-interpretable explanations from agent-based models[17]. In this study, we deploy evolutionary model discovery to investigate the cognitive biases behind the emergence of ideological polarization within a population of agents. An agent-based model of opinion dynamics is developed to simulate a social network, where agents are able to create or sever links with other agents, and update an internal ideological stance based on their exposure to their network neighbors. Cognitive biases grounded in social-psychological theory constitute the agent rules that govern the execution of these actions. These biases, along with operators, are included in the genetic program as terminals to search for agent rules that generate high polarization among the population of agents.

Our findings indicate that having heterogenous rules is more likely to produce polarized social networks. We find that polarized social networks are likely to emerge when a species executing confirmation bias coexists with a species executing either egocentric bias, attentional bias, or cognitive dissonance.

2 RELATED WORK

Agent-based models are a widely utilized tool by the computational social science community for explaining macro-scale social phenomena as a result of their individual-scale behaviours [33]. Mimicking real-world complex adaptive systems, agent-based models describe the individual-scale mechanisms which drive interactions between agents. This *from the bottom up* approach has the advantage of capturing the amplification of behaviors that may seem simple at the individual scale, but can generate unpredicted, macro-scale outcomes [11]. Agent-based models are, therefore, able to capture the path dependency of complex social phenomena by following the interaction pathways that individuals in the real-world undergo [8]. A classic example is the emergence of strongly segregated neighborhoods in populations that accept more than half their neighbors to be of an opposite race [30]. Techniques such as parameter optimization and rule induction have been used to extract theory interpretable findings from agent-based models [28]. Furthermore, frameworks such as Agent Zero demonstrate how the individual-scale decision-making process can be made transparent and constitute of theory-defined human-interpretable units [10].

In particular, genetic programming [22, 24] has proven to be a useful explainable artificial intelligence tool to help discover human-interpretable explanations of social phenomena in an approach known as *inverse generative social science* [1, 33]. Greig and Arranz construct a domain specific language for a genetic program in order to evolve flocking behaviors in the Boids model and formulae for adopting opinions in an opinion dynamics model from a large set of basic mathematical operators and problem specific variables [16]. While this approach allows for a larger search space for possibly fitter rules, it can be challenging to reinterpret the evolved rules into behaviors that are documented in the respective domain. Vu et al, used multi-objective grammar-based approaches [31, 32] to identify factors behind alcohol use starting with existing normative and theory based factors to generate new hybrid models that outperform existing models in the literature. Gunaratne et al. have developed the evolutionary model disocvery framework, which combines genetic programming and random forest feature importance analysis to quantify importance of hypothesized explainble units of behavior towards the emergence of complex social phenomena. The framework has been used to identify factors driving socio-agricultural behavior of an ancient civilization [19, 20], factors driving residential segregation [17], and factors driving response to information overload on social media [21].

3 METHODOLOGY

In order to investigate factors driving ideological polarization, an agent-based, opinion dynamics model was developed, representing a social network of individuals adjusting their stance on a topic as influenced by their neighbors and internal biases. The agents of this model were arranged into the nodes of a scale-free network of 200 nodes with a minimum degree of 1, generated by a preferential attachment algorithm [2]. Each agent had a stance state, s, where $s \in [-1, 1]$, with -1 and $+1$ being extreme stances and 0 being neutral. At each simulation step, each agent, $m \in M$, of the population, M, decided whether to update their stance state, m_s, according to a rule, u. The agent's rule, m_u, was defined by a collection of cognitive biases. At each simulation step, m, would decide to randomly select one of the biases in m_u and execute the actions related to the selected cognitive bias. These actions determined whether the agent would create links with a new neighbor, sever connections with an existing neighbor, or adopt a new belief state based on the belief states of its neighbors. The agent-based model was implemented in NetLogo 6.2.0 [35, 36].

Seven cognitive biases where hypothesized to have an effect on ideological polarization and are summarized below along with descriptions of their implementation.

Anchoring Bias is the tendency to side with information initially received [12]. Accordingly, this factor was implemented in the agent-based model to make the executing agent set another, randomly selected, neighboring agent as an *anchor* and executions of this factor would cause the anchor's stance to be reported. If the link with the anchor were to be severed, a new anchor would be chosen in the same fashion.

Attentional Bias is the tendency to focus on particular stimuli while ignoring others [5, 37]. This factor was implemented to report the stance of the neighboring agent that had the least difference from the stance of the executing agent.

Cognitive Dissonance is when a person experiences psychological stress when exposed to those with beliefs, ideas, or values contradictory to their own. Individuals experiencing cognitive dissonance will do all in their power to change the situation [6]. Cognitive dissonance has been investigated previously in multi-agent models of polarized opinion dynamics [3]. Opinion dynamics models driven by cognitive dissonance has been shown to have influence on network structure and result in polarized and clustered networks with low opinion diversity [34]. Cognitive dissonance was implemented such that agents executing this bias would select and sever links with, out of all neighboring agents with stance value of opposite polarity, the agent with the maximum difference in stance. The executing agent's own stance would then be reported.

Conformity Bias is the tendency to adopt the most popular stance as perceived. Individuals experiencing conformity bias attempt to match their own attitudes, beliefs, and behaviors to group norms [9]. Accordingly, this bias is implemented such that the factor would report the most popular stance from those stances of the executing agent's neighbors.

Confirmation Bias is the tendency of individuals to specifically search, favor, interpret, or recall information in a manner that confirms one's own beliefs or values, while excluding any evidence to the contrary [25]. This bias is implemented such that the executing agent would search for the neighbor of their current neighbors that has the closest stance to their own and create a link with them. The new neighbor's stance would then be reported.

Egocentric Bias Is the tendency to believe that one's beliefs, actions, or perspectives are known and accepted by others [15, 27, 29]. This bias was implemented such that the factor would report executing agent's own stance.

Recency Bias Is the tendency to rely more heavily on recent observations and experiences than older ones [4, 14]. This factor was implemented such that the executing agent would create a link with a randomly selected agent, while severing the oldest link in their local neighborhood. The factor would then report the new neighbor's stance.

In addition to NetLogo procedures for the above cognitive biases, operators to 1) append two factors into a list (*enlist*), 2) select a random factor from a list (*pick_from_list*), and 3) speciate rule execution (*exec_two_species* and *exec_homogenous*), were included in the genetic program.

The degree of polarization, ϕ, was measured as the primary metric of simulation fitness. Given the mean stance of the population of agents is $\tilde{s}$, ϕ was calculated as the root mean squared deviation of all agent stances from $\tilde{s}$ as shown in eq. 1.

$$\phi = \sqrt{\frac{\sum_{m \in M}(\tilde{s} - m_s)}{|M|}} \tag{1}$$

Evolutionary model discovery [17, 18], an open-source python tool for identifying factors driving macro-scale complex adaptive

Primitive	Return Type	Parameter Types
anchoring_bias attentional_bias cognitive_dissonance conformity_bias confirmation_bias egocentric_bias recency_bias	stance_value	-
exec_two_species	stance_operation	[stance-value, stance_value]
exec_homogenous	stance_operation	stance_value
enlist	stance_values	stance_value
pick_from_list	stance_value	stance_values

Table 1: Details of the meta-return types and meta-parameter types of the factors and operators included in the genetic program.

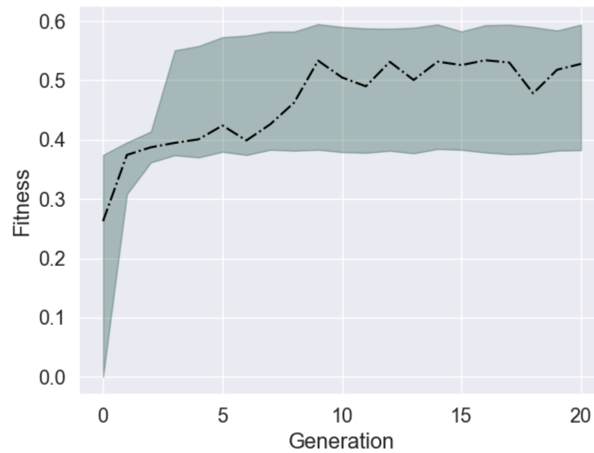

Figure 1: Median and inter-quartile range of fitness of all 20 genetic program runs over 20 generations. Fitness is measured as the polarization intensity, $\phi \in [0, 1]$. Most runs converge to a fitness $0.4 \lessapprox \phi < 0.6$.

system phenomena in NetLogo agent-based models, was executed on the opinion dynamics model. Evolutionary model discovery functions in two steps: 1) Factors hypothesized to generate the target macro-phenomena are included in a genetic program, which automatically programs NetLogo models with rules that are increasingly representative of the target outcome; and 2) A random forest regressor is trained on the factor presence to fitness data generated by the genetic program and used to estimate the permutation accuracy importance of the hypothesized factors. Evolutionary model discovery accepts NetLogo procedures as either factors, procedures that have direct explanatory value and must be included in the importance analysis, and operators, procedures that are do not help directly explain the target outcome, but are required to construct agent rules. Both factors and operators are included as primitives in the genetic program. Evolutionary model discovery is able to identify tags in the NetLogo source code comments that inform which procedures are factors and which are operators, alongside certain metadata, such as return types, parameter types, and polarity of operators. These comment tags are also used to identify the point in the NetLogo model where the agent rule is to be written to by the genetic program. NetLogo procedures are not strongly typed but evolutionary model discovery allows the user to specify meta-types for each factor or operator that can be used to control which procedures the genetic program can pass to into another. Evolutionary model discovery parses the procedures and their associated metadata to generate primitives for the genetic program as Python functions. A summary of the primitives used along with their meta-return types and meta-parameter types are provided in 1. The genetic program then generates rules based on the provided primitives. Each genetic program individual encodes a rule in a tree representation of the parsed factors and operators. One-point crossover and point mutation are used with tournament selection. The genetic program in the evolutionary model discovery tool is implemented using the DEAP toolkit for evolutionary algorithms [13].

The hyperparameters of the genetic program are as follows. Point mutation is applied at a rate of 0.05, single-point crossover is used at a rate of 0.8. Parent selection is performed with tournament selection of size 3. The population size is set to 20 individuals and run for 20 generations. The minimum tree depth of each individual is set to 3, and the maximum set to 20. 10 replications of the genetic program are performed.

Evolutionary model discovery compiles and simplifies the generated rules to identify the coefficients of the factors in each of the genetic program individuals, which is referred to the *presence* of the factor. In this case, there are no negation operators, so the presence of factors is the number of occurrences of the factor within the simplified rule.

The second step of evolutionary model discovery is the factor importance analysis. A random forest regressor [7] was trained on the data produced by the genetic program to predict the fitness of individuals in the genetic program. The input features used by the random forest are the presence values for each factor, in this case a total of 7 features. Evolutionary model discovery used scikit-learn's implementation of the CART algorithm for training a random forest of 500 trees [23, 26]. The trained random forest was then used for permutation accuracy importance analysis to quantify the importance of each factor in the prediction of the fitness of the genetic program individual it appeared in.

Finally, insights from the factor importance analysis were used to construct a set of rules that were able to produce social networks with a high degree of ϕ.

4 RESULTS

Overall, we observe that the genetic program converges towards rules that produce high polarization within approximately 9 generations. Fig. 1 displays the median and inter-quartile range of ϕ over all 20 genetic program runs over 20 generations. Most of the runs achieve polarization intensities around $\phi \approx 0.5$, by generation 9, after which the inter-quartile range remains at $0.4 \lessapprox \phi < 0.6$.

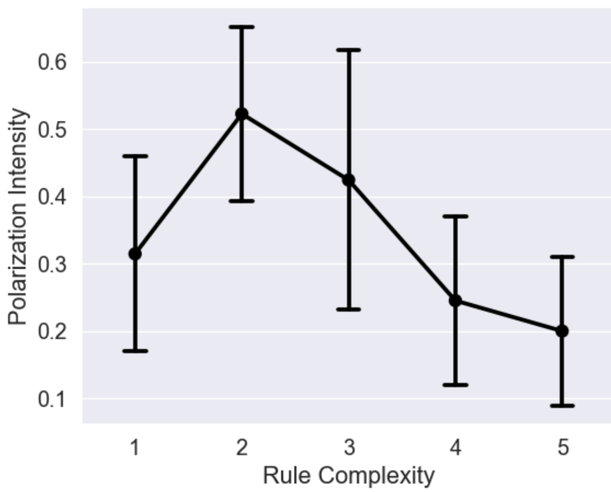

Figure 2: The mean (points) and standard deviations (error bars) of polarization intensity against rule complexity.

Fig. 2 plots the ϕ of simulations against the complexity of the rule. Rule complexity is measured as the number of unique cognitive biases used in rules in the entire population, regardless of speciation. It can be seen that ϕ is highest when rule complexity is moderate, around 2. Higher rule complexity, i.e. beyond 3, generally leads to lower ϕ.

Fig. 3 compares the absolute rule length, i.e., the absolute count of all cognitive bias primitives in each rule regardless of speciation, against rule complexity and ϕ. There is a strong positive correlation between rule length and rule complexity, which saturates around a rule length of 5. In other words, rules longer than 5 generally have the same complexity and never the full set of cognitive biases. Also, ϕ is highest for rules that are between lengths 2 and 5 for complexities 2 and 3. The low fitness of rules longer than 5 seems to indicate instances where bloat in the genetic program individuals may have occurred.

Fig. 4 illustrates the selection pressure on each cognitive bias as the genetic program progresses. Shown are the mean presence of each congitive bias for all runs by generation, along with colors indicating mean ϕ of the rules that contain them. It is clear that anchoring bias, conformity bias, and recency bias are quickly selected out of the population. Cognitive dissonance has a slower, unsteady decline in selection. Attentional bias has high presence throughout the genetic program runs, with a slight decline after the 10^{th} generation. In contrast, confirmation bias and egocentric bias both have monotonically increasing mean presence, ending with high presence at the termination of the genetic program runs.

Tab. 2 lists the top 20 rules produced by the genetic program that produce the highest ϕ. All 20 rules are of two species and one species is always exclusively driven by confirmation bias. Most of the second species in these rules are driven by egocentric bias while 3 are driven by attentional bias.

Fig. 5 displays the results of the permutation accuracy importance of the random forest regressor trained on the factor presence

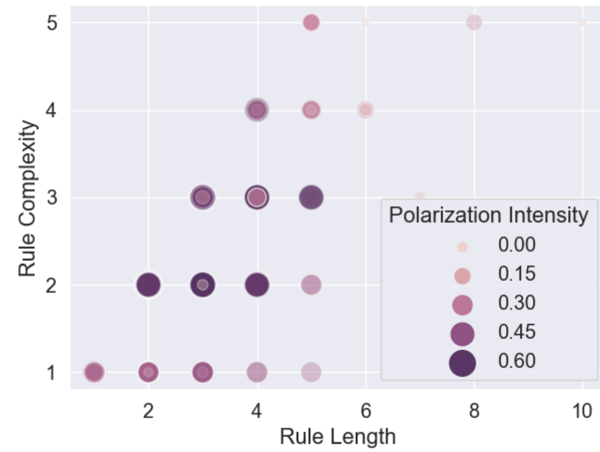

Figure 3: Rule complexity by rule length and corresponding polarization intensity produced by simulation.

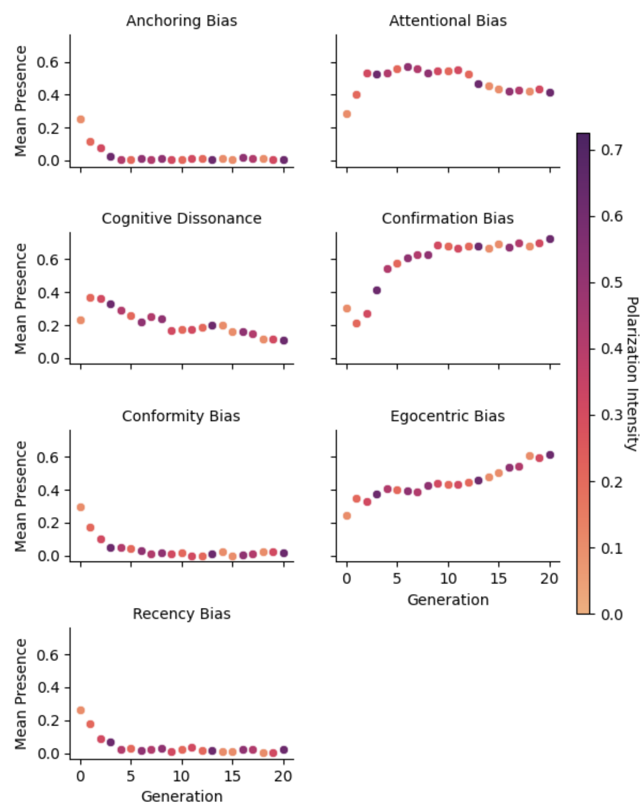

Figure 4: The mean presence and polarization intensity of rules containing each factor for all genetic program runs over generations.

Rule	Polarization Intensity
exec-two-species ([egocentric-bias], [confirmation-bias])	0.725554
exec-two-species ([egocentric-bias], [confirmation-bias])	0.725554
exec-two-species ([confirmation-bias], [egocentric-bias])	0.723651
exec-two-species ([confirmation-bias], [egocentric-bias])	0.722858
exec-two-species ([attentional-bias], [confirmation-bias])	0.718315
exec-two-species ([egocentric-bias], [confirmation-bias])	0.715486
exec-two-species ([confirmation-bias], [attentional-bias])	0.712952
exec-two-species ([confirmation-bias], [egocentric-bias])	0.710315
exec-two-species ([confirmation-bias], [attentional-bias])	0.707573
exec-two-species ([confirmation-bias], [egocentric-bias])	0.701525
exec-two-species ([egocentric-bias], [confirmation-bias])	0.698011
exec-two-species ([egocentric-bias], [confirmation-bias])	0.694072
exec-two-species ([egocentric-bias], [confirmation-bias])	0.692685
exec-two-species ([egocentric-bias], [confirmation-bias])	0.690823
exec-two-species ([confirmation-bias], [egocentric-bias])	0.690545
exec-two-species ([confirmation-bias], [egocentric-bias])	0.690378
exec-two-species ([egocentric-bias], [confirmation-bias])	0.689613
exec-two-species ([egocentric-bias], [confirmation-bias])	0.688828
exec-two-species ([egocentric-bias], [confirmation-bias])	0.687028
exec-two-species ([confirmation-bias], [egocentric-bias])	0.686380

Table 2: Rules generated by the genetic program of the simulations producing the top 20 polarization intensities.

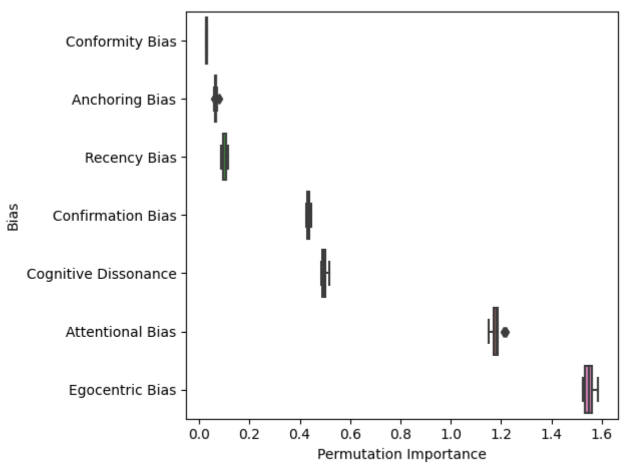

Figure 5: Permutation accuracy importance of the 7 hypothesized cognitive biases towards the prediction of fitness of the genetic program individuals based on the presence of each bias in an agent rule.

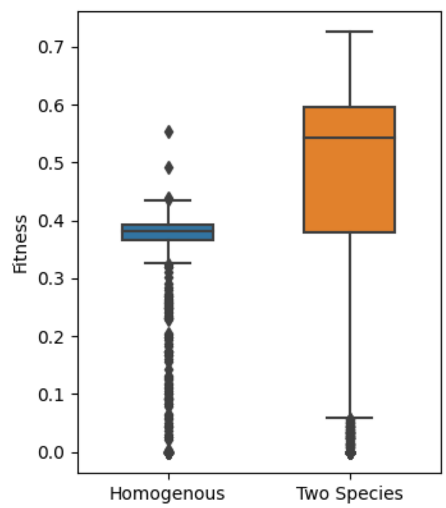

Figure 6: Comparison of marginal fitness under a single rule for the entire population (homogenous) versus two species of rule.

to fitness data. Egocentric bias is seen to be the highest contributing factor towards prediction of the rule fitness, ϕ. Attentional bias is the second most important factor. Cognitive dissonance and confirmation bias show much lower permutation accuracy importance, almost 4 times less important than egocentric bias. In comparison, recency bias, anchoring bias, and confirmity bias have very low importance towards the prediction of ϕ, all more than 16 times less important than egocentric bias.

Next, we compared the marginal fitness of individuals with a single species of rule, i.e., homogenous populations, versus those with two species, heterogenous populations. Fig. 6 compares marginal fitness under homogenous and heterogenous populations. It can be seen that having two species of rules greatly improves the likelihood of generating highly polarized populations.

Using the above insights we construct 4 rules with the 4 cognitive biases that showed highest importance and selection pressure.

The rules were speciated and one species was always driven by confirmation bias as seen the best performing rules. Fig. 7 compares 100 simulation runs of rules constructed using the above insights.

Fig. 8 illustrates two example networks produced by the rules *exec_two_species confirmation_bias], [egocentric_bias])* and *exec_two_species ([confirmation_bias], [cognitive_dissonance]).* Subplots a) and b) are color coded with the stance of each agent, while subplot c) and d) are color coded with the species of rule of each agent. Under both rules, there is a reasonable degree of ideological polarization. It is interesting to see that the network of the rule with the cognitive dissonance had cleaved into two disconnected sub-graphs with opposing polarity. Subplots c) and d) demonstrate that agents of the confirmation bias species were likely to make up the boundary of the social network; taking highly polarized stances when paired with egocentric biased agents and taking both extreme and neutral stances when paired with agents exhibiting cognitive dissonance. Despite the separation of the graph into two sub-graphs confirmation bias paired with egocentric bias is still more likely to produce higher ϕ overall, as all of the confirmation biased agents take more extreme stances in this scenario, while some take neutral stances with the latter rule. This is further illustrated in Fig. 9, which plots histograms of stance under both rules.

5 DISCUSSION

In this study we have utilized genetic programming and random forests to investigate the 7 cognitive biases, derived from theory, with regards to their importance toward the generation of ideologically polarized social networks. An opinion dynamics model was implemented, where agents were arranged in a scale-free network and allowed to interact with their neighbors. Agents were only exposed to information from their network neighbors, and each agent updated a local stance state, and created or severed links to other agents based on a decision rule. This rule consisted of a selection of mechanistic implementations of the 7 cognitive biases and at each simulation step, each agent would select and execute one of their biases. We utilized an open-source tool, evolutionary model discovery, that combines genetic programming and random forests to automatically generate and evolve toward agent rules that generated higher polarization, and compare the relative importance of each cognitive bias toward higher polarization. We also included the option for the agent-based model to have either a single species of rule or two species of rules.

Retrospective analysis of the genetic program results indicate that having two species of rules, greatly increased the likelihood of producing higher polarization. In particular, we observe that the most highly polarized populations consist of two species, with one species driven by confirmation bias and the other with either egocentric bias, attentional bias, cognitive dissonance or a combination of these factors. Further analysis of the generated networks leads us to conclude that individuals driven by confirmation bias are susceptible to adopting extreme stances when exposed to *attentionally biased* or egocentric individuals, who act as anchor points, driving the first species toward more extreme stances. An additional observation was that a single social network may split into two sub-graphs if individuals driven by confirmation bias are exposed to those driven by cognitive dissonance. Finally, conformity bias, anchoring bias, and recency bias showed little to no effect on polarization.

6 CONCLUSION

We investigated the importance of 7 cognitive biases regarding their contribution to the emergence of ideological polarization in social networks. We utilized evolutionary model discovery, an approach that combines genetic programming and random forests, on a model of opinion dynamics. Our results indicate that heterogeneous populations are more likely to produce polarized social networks and that populations where a species with confirmation bias coexists with a species of egocentric bias, attentional bias, or cognitive dissonance was highly likely to experience a higher degree of ideological polarization. In future work, we intend to validate the robustness and generalizability of our findings using real-world survey data.

7 ACKNOWLEDGMENTS

REFERENCES

[1] 2021. Inverse generative social science workshop 2021. https://www.igss-workshop.org/

[2] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74, 1 (2002), 47.

[3] Victor Amelkin, Francesco Bullo, and Ambuj K Singh. 2017. Polar opinion dynamics in social networks. *IEEE Trans. Automat. Control* 62, 11 (2017), 5650–5665.

[4] Vicky Arnold, Philip A Collier, Stewart A Leech, and Steve G Sutton. 2000. The effect of experience and complexity on order and recency bias in decision making by professional accountants. *Accounting & Finance* 40, 2 (2000), 109–134.

[5] Omer Azriel and Yair Bar-Haim. 2020. Attention bias. (2020).

[6] Edwin G Boring. 1964. Cognitive Dissonance: Its Use in Science: A scientist, like any other human being, frequently holds views that are inconsistent with one another. *Science* 145, 3633 (1964), 680–685.

[7] Leo Breiman. 2002. Manual on setting up, using, and understanding random forests v3. 1. *Statistics Department University of California Berkeley, CA, USA* 1 (2002).

[8] Daniel G Brown, Scott Page, Rick Riolo, Moira Zellner, and William Rand. 2005. Path dependence and the validation of agent-based spatial models of land use. *International Journal of Geographical Information Science* 19, 2 (2005), 153–174.

[9] Robert B Cialdini and Noah J Goldstein. 2004. Social influence: Compliance and conformity. *Annu. Rev. Psychol.* 55 (2004), 591–621.

[10] Joshua M Epstein. 2013. *Agent Zero : toward Neurocognitive Foundations for Generative Social Science.* Princeton, NJ : Princeton University Press.

[11] Joshua M Epstein and Robert Axtell. 1996. *Growing artificial societies: social science from the bottom up.* Brookings Institution Press.

[12] Güneş Ertan, Michael D Siciliano, and Deniz Yenigün. 2019. Perception accuracy, biases and path dependency in longitudinal social networks. *PloS one* 14, 6 (2019), e0218607.

[13] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research* 13 (jul 2012), 2171–2175.

[14] Drew Fudenberg and David K Levine. 2014. Recency, consistent learning, and Nash equilibrium. *Proceedings of the National Academy of Sciences* 111, Supplement 3 (2014), 10826–10829.

[15] Thomas Gilovich, Victoria Husted Medvec, and Kenneth Savitsky. 2000. The spotlight effect in social judgment: an egocentric bias in estimates of the salience of one's own actions and appearance. *Journal of personality and social psychology* 78, 2 (2000), 211.

[16] Rory Greig and Jordi Arranz. 2021. Generating Agent Based Models From Scratch With Genetic Programming. In *ALIFE 2021: The 2021 Conference on Artificial Life.* MIT Press.

[17] Chathika Gunaratne. 2019. *Evolutionary model discovery: Automating causal inference for generative models of human social behavior.* Ph. D. Dissertation. University of Central Florida.

[18] Chathika Gunaratne. 2019. Evolutionary Model Discovery Documentation. https://evolutionarymodeldiscovery.readthedocs.io/en/latest/

[19] Chathika Gunaratne and Ivan Garibay. 2017. Alternate social theory discovery using genetic programming: towards better understanding the artificial anasazi.

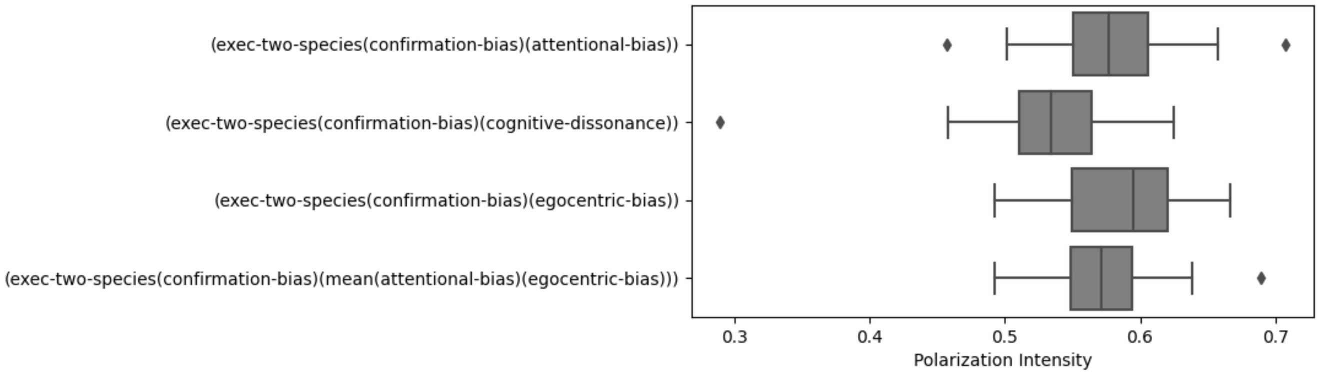

Figure 7: Comparison of polarization produced under 100 runs of rules derived from above insights.

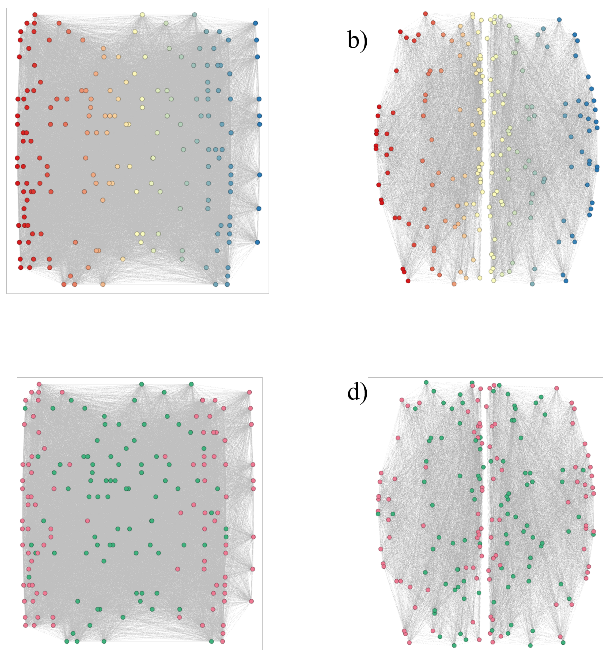

Figure 8: Example illustrations of networks produced under the rules *exec_two_species confirmation_bias], [egocentric_bias])* and *exec_two_species ([confirmation_bias], [cognitive_dissonance])* ((b) and d)). In all four figures agents are arranged horizontally according to stance with neutral agents in the middle and agents to the left and right having increasingly extreme stances approaching -1 and 1, respectively. Subplots a) and b) are colored with the stance of each agent, red corresponding to negative stance, yellow to neutral, and blue to positive. Subplots c) and d) are colored according to the rule species being simulated; in c) green depicts egocentric bias and red depicts confirmation bias, while in d) green depicts cognitive dissonance and red depicts ,confirmation bias.

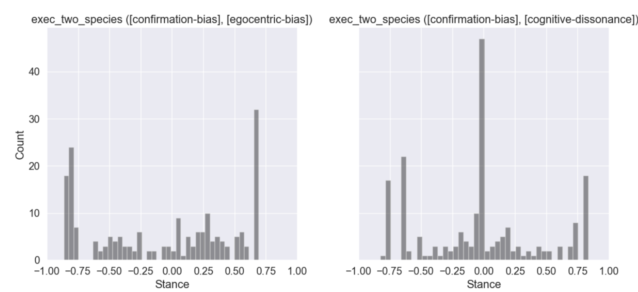

Figure 9: Histograms of stance under the rules-*exec_two_species confirmation_bias], [egocentric_bias])* and *exec_two_species ([confirmation_bias], [cognitive_dissonance])*

In *Proceedings of the genetic and evolutionary computation conference.* 115–122.
[20] Chathika Gunaratne and Ivan Garibay. 2020. Evolutionary model discovery of causal factors behind the socio-agricultural behavior of the ancestral Pueblo. *Plos one* 15, 12 (2020), e0239922.
[21] Chathika Gunaratne, William Rand, and Ivan Garibay. 2021. Inferring mechanisms of response prioritization on social media under information overload. *Scientific reports* 11, 1 (2021), 1–12.
[22] John R Koza. 1992. *Genetic programming: on the programming of computers by means of natural selection.* Vol. 1. MIT press.
[23] Gilles Louppe. 2014. *Understanding Random Forests: From Theory to Practice.* Ph. D. Dissertation. Université de Liège, Liège, Belgique. Chapter 6.1.2: Importances in forests.
[24] James McDermott and Una-May O'Reilly. 2015. Genetic programming. In *Springer Handbook of Computational Intelligence.* Springer, 845–869.
[25] Raymond S Nickerson. 1998. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology* 2, 2 (1998), 175–220.
[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[27] Uwe Peters. 2016. Human thinking, shared intentionality, and egocentric biases. *Biology & Philosophy* 31, 2 (2016), 299–312.
[28] William Rand. 2019. Theory-interpretable, data-driven agent-based modeling. *Social-behavioral modeling for complex systems* (2019), 337–357.
[29] Michael Ross and Fiore Sicoly. 1979. Egocentric biases in availability and attribution. *Journal of personality and social psychology* 37, 3 (1979), 322.
[30] Thomas C Schelling. 1971. Dynamic models of segregation. *Journal of mathematical sociology* 1, 2 (1971), 143–186.
[31] Tuong M Vu, Charlotte Buckley, Hao Bai, Alexandra Nielsen, Charlotte Probst, Alan Brennan, Paul Shuper, Mark Strong, and Robin C Purshouse. 2020. Multiobjective genetic programming can improve the explanatory capabilities of mechanism-based models of social systems. *Complexity* 2020 (2020).

[32] Tuong Manh Vu, Eli Davies, Charlotte Buckley, Alan Brennan, and Robin C Purshouse. 2021. Using multi-objective grammar-based genetic programming to integrate multiple social theories in agent-based modeling. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 721–733.

[33] Tuong Manh Vu, Charlotte Probst, Joshua M Epstein, Alan Brennan, Mark Strong, and Robin C Purshouse. 2019. Toward inverse generative social science using multi-objective genetic programming. In *Proceedings of the genetic and evolutionary computation conference*. 1356–1363.

[34] Roger M Whitaker, Gualtiero B Colombo, Liam Turner, Yarrow Dunham, Darren K Doyle, Eilish M Roy, and Cheryl A Giammanco. 2021. The coevolution of social networks and cognitive dissonance. *IEEE Transactions on Computational Social Systems* (2021).

[35] Uri Wilensky. 1999. *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University. http://ccl.northwestern.edu/netlogo/

[36] Uri Wilensky and William Rand. 2015. *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. MIT Press.

[37] Ariel Zvielli, Amit Bernstein, and Ernst HW Koster. 2015. Temporal dynamics of attentional bias. *Clinical Psychological Science* 3, 5 (2015), 772–788.