Ethnocentrism Model Construction Proposal

Lihang Tian 975404
Mayank Sharma 936970
Yuanlong Zhang 772312

Abstract

This proposal describes the model construction process for Ethnocentrism with a brief introduction of the previous research about Ethnocentrism, the existing Ethnocentrism model in NetLogo, the design of our model, potential experiments as well as time schedule and task distribution.

Research objectives

Model Summary

This agent-based model is implemented according to the research of the “Robert Axelrod and Ross A. Hammond” and traits the interactions via the one-move Prisoner’s Dilemmas.

There are 4 different types of agents used each in the model with 3 different traits. Agents can either fully cooperate, cooperate with the same colour, cooperate with different colour, or not cooperate at all.

The results of the model demonstrate the ethnocentrism could be evolved with the high cost of the corporation and minimal cognitive requirements as well as the absence of other complex social mechanisms.

Purpose of duplicating this model

Corporation is a wired topic in our society. It happens everywhere among all groups from, human to animals. But why do we want to cooperate if there is a cost? The purpose of this model is to investigate how ethnocentric behaviour can be evolved by using a faster computational model which can simulate this behaviour.

In NetLogo's model, we are trying to understand the ethnocentric behaviour among the population of individuals and how different factors can affect the ethnocentric behaviour of the population. The factors include the corporation cost, the cognitive requirements and the absence of other complex social mechanisms.

Users

This study is for anyone who is interested in understanding the evolution of Ethnocentrism and who would download the code to test it.

Researched Model

Existing model description

Agents populate a toroidal lattice that is default set to consisting of 50x50 patches, each with room for at most one individual. The lattice is empty in the beginning, becoming populated through immigration and reproduction as described below. The simulation runs for 2,000 rounds and there are four stages in one round:
Immigration: An agent with random traits enter the population at a random empty site. Each agent has the initial potential to reproduce (PTR) - 0.12
Interaction: Each immigrant interacts once with each of its immediate horizontal and vertical neighbours in a prisoners' dilemma. The cost of giving is 0.01 and the gain of receiving is 0.03.
Reproduction: Each agent is randomly chosen to produce one offspring with the probability of its PTR onto an empty neighbouring patch. If the surrounding positions of the agent are all populated, this agent cannot reproduce in the current round. The offspring may have the same traits as its parent but also has the possibility of 0.005 to mutate.
Death: Each agent has a possibility of 0.1 to die to make room for offspring.

We start the run with SETUP EMPTY. There are multiple factors that can be changed such as COST-OF-GIVING (shows the cost required to cooperate with another agent), COST-OF-RECEIVING (shows the cost gained when another agent cooperates with them), IMMIGRANT-CHANCE-COOPERATE-WITH-SAME (shows the probability that an agent will cooperate with the same agent) and IMMIGRANT-CHANCE-COOPERATE-WITH-DIFFERENT (shows the probability that an agent will cooperate with a different agent). The model plots the STRATEGY COUNTS to track the number of agents using a particular strategy.

By adjusting different variables in the model, will have different results.
Certain outstanding behaviours that happen under certain setting: (Keep updating)

1. When cost of giving < gain of receiving -> Corporate with same colour took the lead in long term
2. When cost of giving > gain of receiving -> corporate with no one strategy is dominant

Assumptions:
The agents can have their offsprings, who will inherit the agent's colour and strategy. Which allows for certain strategies to dominate over time.
Also, the offsprings have a chance to mutate and the dominant can be replaced with a more successful strategy when the environment changes.
To establish fairness, it is assumed that all agents are in random order.

States

In the model, each agent will have three traits: 1, colour 2, whether they cooperate with same coloured agents 3, whether they cooperate with different agents.

Using different settings on traits 2 & 3, there will always be four different agent strategies: 1, always cooperate 2, always not corporate 3, only corporate with same coloured agents 4, only corporate with different coloured agents

In each event time, an agent could be in one of possible 4 states: 1: corporate with same coloured agent 2, corporate with different coloured agent 3, refuse to corporate 4, died

Update rules

Each agent appears at a random location based on the number of IMMIGRANTS-PER-DAY with random traits. Agents start with the property INITIAL-PTR chance of reproducing. Each adjacent pair tries to reproduce based on the prisoner's dilemma. Randomly, each agent is given a chance to reproduce and a MUTATION-RATE is applied to indicate how much the offspring has mutated. In the end, the DEATH-RATE decides the probability of an agent dying freeing space for other agents and offsprings.

The Team's Model Design

Abstract summary

The initial model is based on the description of Ethnocentrism model in "Robert Axelrod and Ross A. Hammond". The most basic moving unit, "Agent", has some properties and strategies that it uses to interact with different agents. The "Simulator" controls the agents based on some constant variables. A final statistical data of one simulation can be extracted from "Board".

The definition of Classes

Class Name: Agent

Attributes:

- Color [Enum]
- Coordinates [int][int]
- CoopWithSame[boolean]
- CoopWithDifferent [boolean]
- LifePoint [double]

- Shape [Enum]

Methods:

- Agent()
- Getters
- Setters

Class Name: Simulator

Attributes:

- COST_GIVING [double]
- GAIN_RECEIVING [double]
- DEATH_RATE [double]
- MUTATION_RATE [double]
- INITIAL_PTR [double]
- IMMIGRATION_PER_DAY [double]
- ROUND_NUM [double]
- LATTICE_SIZE [int]
- ----------Future version
- GUI/Command line input
- Unlimited rounds
- More rich agent behaviours
    - Chance_cor_same
    - Chance_cor_other

Methods:

- Main()
- AgentSpamming()
- AgentReproduce()
- AgentInteraction()
- AgentDeath()
- FindSurroundEmptySpace()
- FindNeighbours()

Class Name: Board

Attributes:

- width [int]
- length [int]
- gameBoard [Agent][Agent]

Methods:

- Getters
- Setters
- getAltruistsNumber()
- getEthnocentricsNumber()
- getCosmopolitanNumber()
- getEgoistNumber()

Experiments Design

Experiment 1

Modify the value of one parameter at each time and see the change. Especially pay attention to the change of dominator when one of the parameters is in limited value.
E.g. Modify the values of related parameters together at one time, such as the cost-of-giving and gain-of-receiving, immigrant-chance-corporate-with-same and immigrant-chance-corporate-with-different. And adjust them according to a different combination.

Goal:

To observe the change of the dominator of the world and the curve change trend with the adjustment of different parameters.

Predictions:

Cost-of-giving and gain-of-receiving directly affect the dominator of the world.
If the mutation-rate is too large, the world will be in a balanced state and have no dominator.

Experiment 2

Start with all initial default settings and plot the result. Start with full settings and plot the result. Then select the max ethnocentrism settings according to NetLogo.

Goal:

To check if Ethnocentrism exists under all cases and ethnocentrism should be maximum with all settings set to zero but the cost of giving, immigration per day and immigration chance cooperate with same should be maximum.

Predictions:

The graphs should show that ethnocentrism exists under all cases and according to the specific ethnocentric settings, the graph of ethnocentrism should show maximum.

Schedules & Task Distribution

Date	Tasks

11/05 SAT	Simple version individual parts done
12/05 SUN	Simple version parts merged & debug
13/05 MON	Run initial experiments
14/05 TUE	Analysis results
15/05 WED	Design modification - clear and succinct
15/05 WED	Design modification - potentially extensible
16/05 THU	Modified version individual parts done
17/05 FRI	Modified version parts merged & debug
18/05 SAT	Complete modified version of the model - Executability
19/05 SUN	Complete modified version of the model - Code Formatting
20/05 MON	Modify experiments
20/05 MON	Run modified experiments
21/05 TUE	Analyse modified experiments results
22/05 WED	Report P1- Background & Model
23/05 THU	Report P2- Replication & Extension
24/05 FRI	Report P3- Results & Discussion
25/05 SAT	Report P4- Appendix (team work& challenges- half page)
25/05 SAT	Polish the final reporting (<8 pages || 1500words)
26/05 SUN	Polish the Final Code - scripts that lead to report figure & read me
27/05 MON	Complete Group contribution feedback Package and submit the assignment