**Automating Agent-Based Model Construction using IRL**

**Kamwoo Lee (kl9ch), Mark Rucker (mr2an)**

**Abstract**

This is abstract…

**1 Introduction**

The critical observation of Agent-Based Modeling (ABM) is that simple rules for individuals can result in interesting emergent behavior at an aggregate level. Unfortunately, discovering these simple rules, is often very challenging and requires deep insight about an agent’s behaviors. If we can extract the behavioral rules (i.e., reward functions) of individuals from data through Inverse Reinforcement Learning (IRL), it could be a meaningful contribution to automation of ABM construction.

**1.1 Agent Based Modeling (ABM)**

(General explanation of ABM)

ABM… how system level properties emerge from the adaptive behavior of agents (Railsback 2001) as well as how the system affects individuals.

**1.2 Inverse Reinforcement Learning (IRL)**

To understand Inverse Reinforcement Learning one first needs to understand Reinforcement Learning. Reinforcement Learning is a machine learning framework built to optimize and control MDP’s. Within this framework are five primary components:

1. State – The set of all observations at any given point in an agent’s environment
2. Actions – The set of all actions which an agent is allowed to take at any given time step
3. Transitions – The probabilities of ending up in a State after taking an Action in the current State
4. Policy – A function which maps States to Actions
5. Reward – A function which characterizes

When solving a traditional Reinforcement Learning problem all the above components are known except for the Policy. One then tries to solve for a policy which provides maximum reward. When solving an Inverse Reinforcement Learning problem the same five components are considered but instead of solving for a Policy we are concerned with finding the Reward.

Within the research community Inverse Reinforcement Learning has been studied with varying levels of constraints. In the simplest case, States, Actions, Transitions and an optimal Policy are known fully -- leaving only the Reward as unknown. At the other extreme, often times only the State and Action spaces are fully known. In these instances, in place of Transitions and Policies, a sequence of state action pairs are provided. These sequences are referred to as trajectories and, in the worst cases, it isn’t even assumed that given trajectories are optimally solving for the true reward.

**2 Proposed Method: Automatic ABM Construction Method**

**2.1 Method Flow**

1. Run Inverse Reinforcement Learning (IRL) algorithm to extract agents’ behavioral policies (rewards functions behind their behavior) from a dataset.
2. With a Machine Learning (ML) clustering algorithm, classify the behavioral policies. Alternatively, feature expectations of agents’ behaviors are classified first, and then IRL can be run to obtain classified behavioral policies.
3. Construct ABM with the learned behavioral policies.
4. Incorporate any analysis measures and intervention ideas into the model.

**2.2 Validation method**

Since this is a new method constructing an agent-based model, it needs an additional validation (conceptual validation and operational validation??).

**3 Experiment Setup**

The result of this project will seek to simulate human segregation behavior with ABM. The rules of each individual agent will be extracted from synthetic data using IRL algorithm after we classify their feature expectations by a machine learning clustering algorithm. We expect that the extracted and classified reward functions can provide a newly constructed ABM with rich but concise rules for agents which are often hard to get from observation alone.

**3.1 Synthetic Data**

Using NetLogo, 20 sets of data have been generated for 700 people’s behaviors with regard to conversation and movement. Basically, 700 people are located in a public space, and they talk with others and move continuously. They are distinguished by their innate characteristic with 2 levels. This characteristic can be considered as ethnicity (Black/White), religion (Hindu/Muslim), or any other traits with 2 levels. There are 3 types of people and each person can take one of 4 actions at a time.

* kinds of people: unbiased people, biased people, racists
* possible actions:

1. start a conversation with a nearby person
2. continue a conversation for another tick
3. move short distance with random direction
4. move long distance with random direction

Time in this data is discretized in ticks (time scale in NetLogo). A proximity radius is defined to identify nearby people and distance. Short distance is defined as a length slightly less than the proximity radius and long distance as a length slightly more than the radius.

The three types of people have their own behavioral patterns as follows:

* Unbiased people: Speak to anyone for random duration (1 – 10 ticks), Move either short or long distance randomly
* Biased people: Speak to people with same characteristic for 5 ticks and people with different characteristic for 2 ticks, Move short distance after meeting with a same character and long distance after meeting with a different character.
* Strongly biased people: Speak for 1 tick with a same character and 10 ticks with a different character, Move short distance always.

**3.2 MDP Formalization**

We constructed a Markov Decision Process (MDP) with 4 state variables, 4 actions as shown below.



There are 4 types of illegal actions in this formulation.

1. Continue conversation while not having a conversation

2. Start or continue a conversation when there's no potential partner

3. Start or continue a conversation when it reaches the maximum conversation length

4. Start a conversation when having a conversation

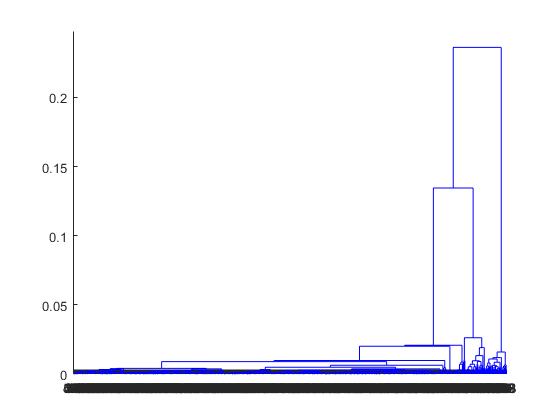
We introduced a “limbo” state which is an absorbing state in order to deal with these illegal actions. Since the IRL iteration process starts with an arbitrary policy, many of feature expectations in an IRL iteration should include the illegal actions. By leading those actions to the absorbing limbo state, we tried to prevent the IRL from learning illegal actions as a result policy.

**3.3 Transition Probability**

The transition probabilities is 3 dimensional matrix with 29 departure states, 29 destination states, and 4 actions. Each element of the matrix has the probability that an agent lands on the destination state when taking an action in the departure state. It is often difficult to define the transition probability for all actions in all states because the transition probability has to be as neutral as possible in order for IRL to learn a specific policy not from the environment but from the trajectories of movement. Another reason of difficulty is that not all actions in all states can be observable despite the fact that a complete set of transition probabilities is necessary to formalize an MDP.

We solve the difficulty in this way. We retrieved average transition probability of each action in each state from the trajectories of the synthetic data. We then select the most typical case as a model probability and project the model probability to actions which have the same structure as the model case. For example, we gave action A1 in state S10~18 (some of them are unobservable in the synthetic data) the transaction probability of action A1 in state S9 since action A1 is structurally same regardless of the conversation length.

**3.4 Clustering**

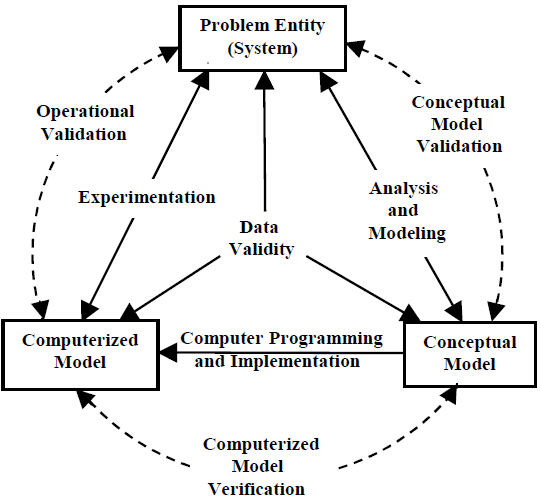
Hierarchical clustering of feature expectation with cosine distance function

**3.5 IRL algorithm and Approximation of Stochastic Policy**

We used Abbeel and Ng’s projection algorithm. This algorithm mixes together output policies to obtain a policy whose feature expectations are most similar to that of the sample data (or expert in IRL parlance). Even though the randomization step selecting between policies should occurs at the start of a trajectory, we tried to mix the result policies on each step to get a single stochastic policy. We believe that this mixture is not a significant breach of assumptions in this experiment because all agents in a cluster have a homogeneous policy and they start their trajectories at the same state in this experiment. Therefore, considering the memoryless property of an MDP, mixing policies at the start of a trajectory is equivalent to mixing policies on each step in this experiment.

**4 Experiment Result**

First thing we conduct is verification and validation of the constructed model. We follow an approach proposed by Sargent (2000).

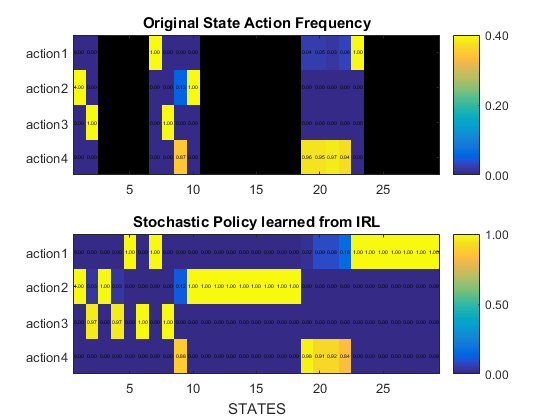


…

…

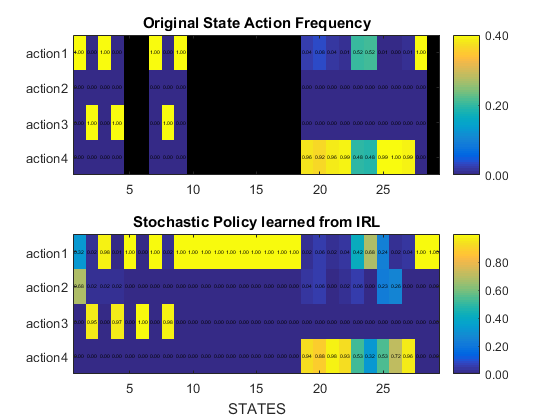
**4.1 Verification**

**Biased people**

****

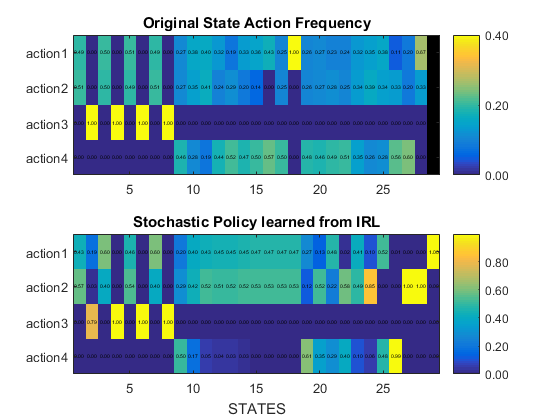
The IRL generates reasonable policies in unobserved states. For example, no action in state S3 [conversation length 0, different recent partner, no potential partner, familiar environment] was observed since biased people always try to go to an unfamiliar environment when the recent partner had a different character. But if a biased person was somehow in state S3, it would take action A2 because making a long move would lead it to state S1 [conversation 0, different recent partner, no potential partner, unfamiliar environment] and makes its feature expectation similar to the expert's feature expectation.

**Racists**

****

We can notice the learned stochastic policy is noticeably different from the action frequency in state S1. That is because when a racist is in state S1 [conversation 0, different recent partner, no potential partner, unfamiliar environment], taking action A2 is almost equivalent to taking action A1 in that racists rarely visit this state and both actions would make its feature expectation equally similar to the expert's expectation.

**Unbiased people**

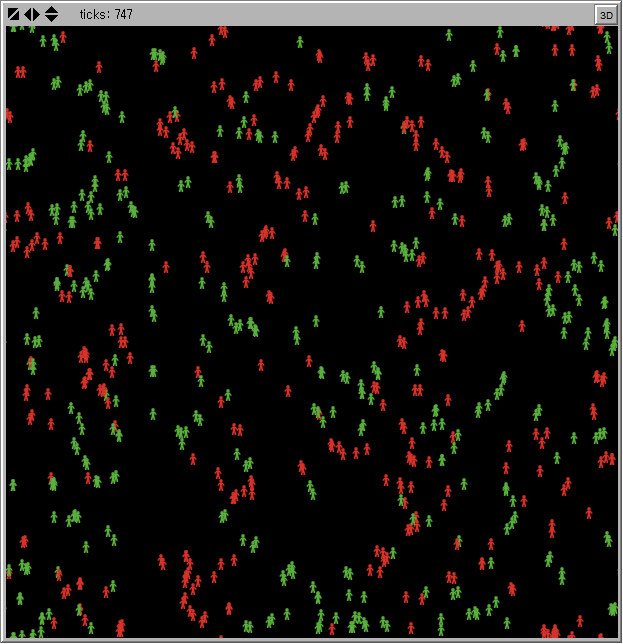
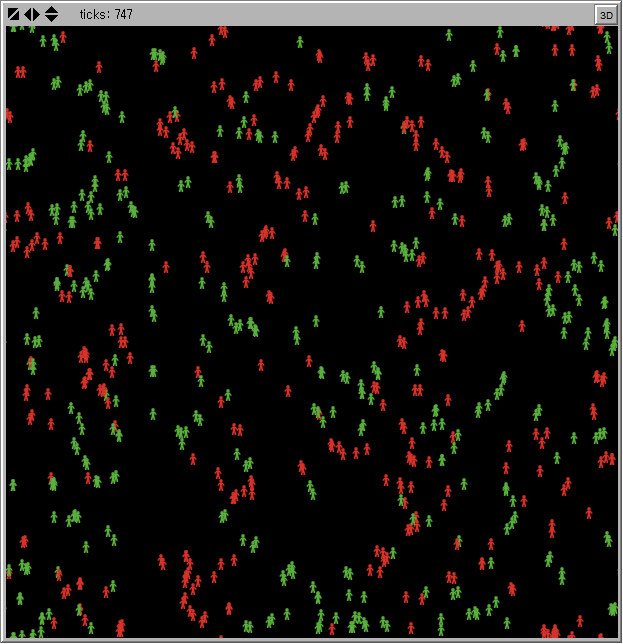
****

The learned policy from IRL shows that what alternative policies can make an agent’s feature expectation similar to the expert's feature expectation. Although it is not an exact copy of the original movement, this learned policy can work well as a rule in an ABM because...

**4.2 Validation**

In order to validate the model, we look at both qualitative and quantitative aspects. We consider that the most important aspect is the aggregated behavior of agents. So, we observe the progress of segregation by visual comparison between the original model (synthetic data) and the newly constructed model. Then, we also observe the result of segregation using the statistical moments (mean and variance) of spatial segregation and social segregation.

Although the whole progress of segregation cannot be presented here, we can show the similarity of segregation after same ticks as shown in figure [ ].



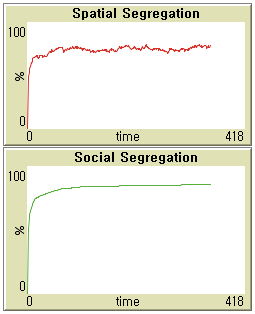
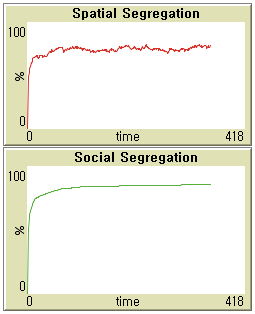
(this will be replaced with the actual results)

For the purpose of statistical validation, we define special segregation and social segregation as follows:

* Spatial Segregation: On average, percentage of majority color agents in a 10-by-10 patch (a proximity radius defined in the synthetic data) on average
* Social Segregation: On average, percentage of conversation lengths with a same character partner.

The mean and variance of them over time are

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| tick interval | | 1-100 | | 101-200 | | 201-300 | | 301-400 | | 401-500 | |
| synthetic data | mean |  |  |  |  |  |  |  |  |  |  |
| variance |  |  |  |  |  |  |  |  |  |  |
| constructed model | mean |  |  |  |  |  |  |  |  |  |  |
| variance |  |  |  |  |  |  |  |  |  |  |



(this will be replaced with the actual results)

**4.3 Example of Policy Simulation**

We present an example of simulation which is based on the behaviors extracted from data using IRL. In this example, we give a network analysis among group of heterogeneous people who have their own rules of conversation and movement.

**5 Conclusion**

**REFERENCES**

Railsback, S.F., 2001. Concepts from complex adaptive systems as a framework for individual-based modeling. Ecol. Model. 139, 47–62.