ODD Framework for an Agent Based Language Shift Model

Tyler Pawlaczyk Dr. Jonathan Martin Dr. Diana White

September 21, 2017

1 Purpose

This model seeks to explain the dynamics of Language Shift in a location, specifically Southern Austria. With little modification from the provided example, uses include simulating the spatial spread of language trends. and any other phenomena that can be postulated as a spatial spread.

2 Entities, State Variables and Scales

2.1 Overview

There is only one acting agent in this model, the LanguageAgent. All LanguageAgents are contained within the same LanguageModel. LanguageModels also contain another singular entity to handle neighborhood calculations, the NeighborList.

2.2 LanguageAgent

The LanguageAgent represents a single location during the duration of the simulation. In all typical uses of this model, the typical state variables will be employed by the LanguageAgents. Extraneous state variables are used in special cases. The extraneous state variables mentioned here are primarily for comparing with results with Prochazaka's 2017 findings.

Many LanguageAgents are contained in a LanguageModel. The neighborhood relation of LanguageAgents is determined by the LanguageModel's NeighborList.

2.2.1 Typical State Variables

- `name` (string)
  Each LanguageAgent contains a string field that is the name of the location it represents.

- `pos` (tuple: (float, float))
  The position of the LanguageAgent. In the provided example, this is the latitude/longitude coordinates of the location. `pos` is used by the NeighborList to determine the LanguageAgent's neighborhood.

- `probability, next_probability` (double)
  The `probability` and `next_probability` store information regarding the probability of speaking either language for the current and next timestep. `next_probability` is required due to the design of the scheduler.

- `population` (integer)
  Each region represented by a LanguageAgent has a population for each timestep. The population is updated every timestep.

2.2.2 Extraneous State Variables

- `p_probability`, `p_next_probability` (double)
  Used to store the probabilities calculated as described in Prochazaka's 2017 paper.

2.2.3 Scale

In the provided example, each LanguageAgent represents a region approximately 1 square kilometer in area.

2.3 LanguageModel

The LanguageModel is the "container" in which the simulations happen. The LanguageModel is responsible for reading in initialization data, initializing LanguageAgent objects, and collecting data.

2.3.1 Typical State Variables

- `diffusion` (list: [float])
  Represents the diffusivity of each language. In the provided simulation, the first element of the list is the German diffusivity, and the second is the Slovene diffusivity. Usually size of two.
- `pop_data` (DataFrame: int)
  Maintains a pandas dataframe of populations for every timestep.
- `neighbors` (NeighborList)
  Maintains a list of neighbors to each agent.
- `agents` (list: [LanguageAgent])
  A list of all agents indexed by their `unique_id`.

2.3.2 Scale

The LanguageModel is scaled for the whole simulation in all times and spaces.

2.4 NeighborList

The NeighborList is responsible initially for generating the nearest neighbors of each agent, and then maintaining a list of neighbors for each agent.

2.4.1 Implementation Details

The NeighborList first obtains the position of all LanguageAgents in the LanguageModel.

3 Process Overview and Scheduling

4 Design Concepts

5 Initialization

6 Input Data

7 Submodels