Isogloss inference

Siva Kalyan and Gereon Kaiping

November 20, 2017

Contents

1 Goal

Given character-state (usu. lexical) data for a set of related languages, we want to able to infer (using either maximum-likelihood or Bayesian methods):

1. the historical network of interaction among the language communities (or among their ancestors);

2. the shape of the isogloss corresponding to each character-state;

3. the sequence of isoglosses for each character.

We would like to model the network of interaction as a *planar graph* (or some subclass thereof), because this seems like a natural way of modeling geographic diffusion. Luckily, there exist efficient algorithms for uniformly sampling the space of planar graphs (or some subclass thereof), e.g. `plantri`. We would like the edges to have costs, to simulate natural (or social) boundaries between communities (which may eventually become boundaries of isoglosses).

Each isogloss is necessarily a connected subgraph of the network. It need not be convex.

2 What we need

2.1 Model for randomly generating isoglosses

Parameters of our model:

1. A planar graph G.

2. A set of costs $\{c_i\}$ associated with each edge (where $0 < c_i < \infty$ for all i).

3. A rate r of population growth.

The instantaneous rate of isogloss creation is given by e^{rt} (where t is time; we scale everything so that $t \in [0, 1]$).

To generate an isogloss, use the following procedure:[1]

1. Pick a node $v \in V(G)$ at random; this will be the "center" of the isogloss.
2. For each neighbor n of v, let $c(v, n)$ be the cost of the edge going from v to n. Then the probability of n joining the isogloss is given by $(e^{-rt})^{c(v,n)}$. Use these probabilities to decide which neighbors get to join the isogloss.
3. Apply the above step for each neighbor of each newly-added node; and keep going until no new nodes are added.

The important thing to note is that each isogloss corresponds to a *subtree* of the network, rooted at the language that serves as the "center".

2.2 Formula for estimating likelihood of an isogloss

Suppose we have an isogloss, and we want to estimate (for a given set of parameters of our model) how likely this isogloss is to arise at a given time t. We do the following:

1. Take all spanning trees of the subgraph defined by the isogloss plus all of its immediate neighbors (considering only trees whose root lies within the isogloss). (Note: This may be a huge number of trees, so in practice we'll have to sample. But there should be good algorithms for uniformly sampling the set of spanning trees of a graph.)
2. For each spanning tree, multiply the probabilities along all edges inside the isogloss with 1 minus the probability of each edge connecting nodes inside and outside the isogloss. (The idea is to calculate the probability that an innovation starting at the root node will diffuse to all the nodes in the isogloss, and *fail* to diffuse any further.) Take the average across all spanning trees.
3. To compute the time-independent likelihood of the isogloss, do the above for each (uniformly-spaced) value of t, and take the average.

2.3 Formula for estimating likelihood of a partially indeterminate isogloss

Take every possible "resolution" of the isogloss, and compute the likelihood as above. Take the average.

2.4 Method for enumerating/sampling from the possible sequences of isoglosses that might give rise to an observed distribution of character states

Once we have this, we can compute the likelihood of the observed distribution of states for a given character. Then it's a short step to implementing maximum-likelihood and Bayesian methods.

[1] This is based on the so-called "Independent Cascade Model" (Goldenberg *et al.* 2001, though they never call it that). The other widely-used model of diffusion on networks is the "Linear Threshold Model"; but this looks more complicated.

Randomly order the character states, and randomly partition $[0, 1]$ into the same number of disjoint intervals. For each character state, if a later character state affects neighboring nodes, then these neighboring nodes should have blanks instead of zeros (since they could in principle have formerly belonged to the older isogloss). Once we have made older isoglosses "indeterminate" in this way, we compute the likelihood of each isogloss as above, taking into account the time interval within which we expect the isogloss to arise.

At this point, we'll want to make the ordering of the states of each character a parameter in our model. I wonder if this is getting too complicated....

Done!