A Framework for Representing

Language Acquisition in a Population Setting

Jordan Kodner
Christopher Cerezo Falco
University of Pennsylvania

ACL - July 16, 2018
Melbourne

Language Change

Languages change over time

- Both an internal and external process
- Fundamentally social
- Individuals acquire language and transmit it to future generations
- New variants propagate through populations

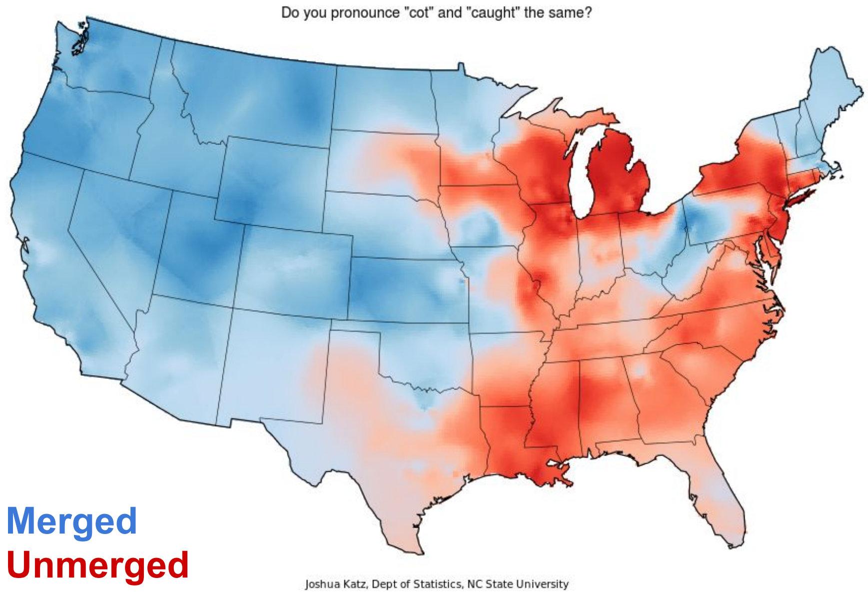
Modelling Change

- Must model how the individual reacts to linguistic input and to the community

Example - The *Cot-Caught* Merger

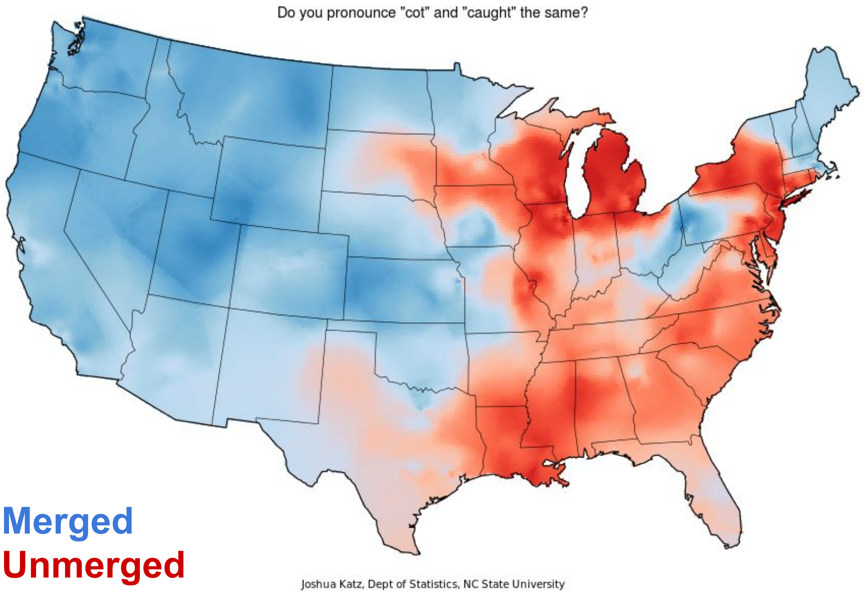
- /ɒ/ "*cot*" is pronounced the same as /ɔ/ "*caught*"
- *Minimal pairs* distinguished by /ɒ/~/ɔ/ become *homophones*

/ɒ/	/ɔ/
cot	*caught*
Don	*Dawn*
collar	*caller*
knotty	*naughty*
odd	*awed*
pond	*pawned*



Example - The *Cot-Caught* Merger

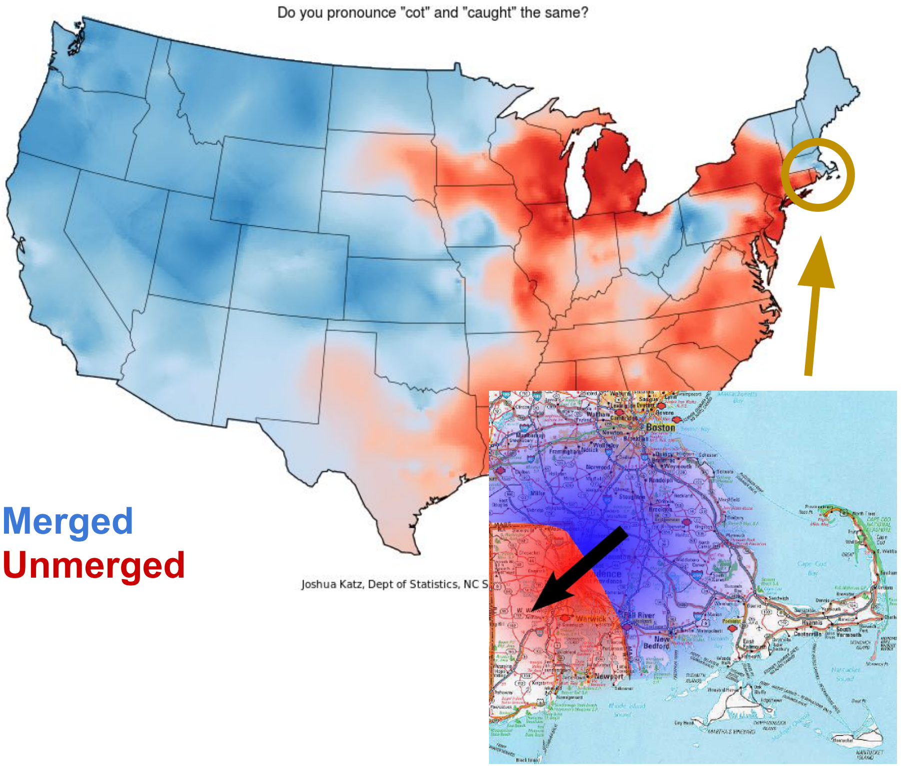
- /ɒ/ "*cot*" is pronounced the same as /ɔ/ "*caught*"
- Present in many dialects of North American English
  - Eastern New England
  - Western Pennsylvania
  - Lower Midwest
  - West
  - Canada (all)



Merged
Unmerged

Example - The *Cot-Caught* Merger

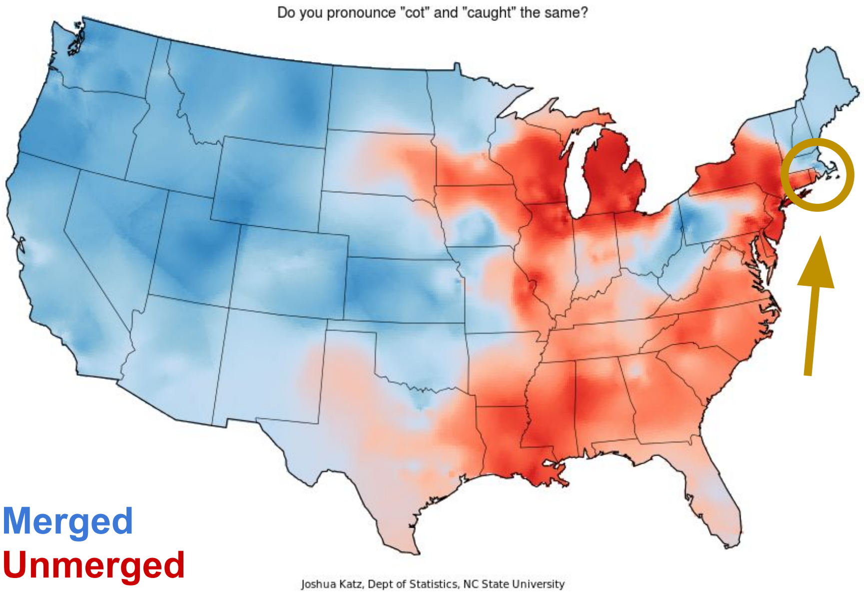
- /ɒ/ “*cot*” is pronounced the same as /ɔ/ “*caught*”
- Present in many dialects of North American English
    - Eastern New England
    - Western Pennsylvania
    - Lower Midwest
    - West
    - Canada (all)
- Spreading into Rhode Island (Johnson 2007)



Merged
Unmerged

Example - The *Cot-Caught* Merger

- /ɒ/ "*cot*" is pronounced the same as /ɔ/ "*caught*"
- Present in many dialects of North American English
  - Eastern New England
  - Western Pennsylvania
  - Lower Midwest
  - West
  - Canada (all)
- Spreading into Rhode Island
- Rapid! Families with Non-merged parents and older siblings but merged younger siblings



Merged
Unmerged

Existing Frameworks

Three Classes of Framework

1. Swarm Frameworks
2. Network Frameworks
3. Algebraic Frameworks

Three Classes of Framework

1. **Swarm Frameworks**
    - Individual agents on a grid moving randomly and interacting (ABM)
    - e.g., Harrison et al. 2002, Satterfield 2001, Schulze et al. 2008, Stanford & Kenny 2013

Three Classes of Framework

1. **Swarm Frameworks**
    - Individual agents on a grid moving randomly and interacting (ABM)
    - e.g., Harrison et al. 2002, Satterfield 2001, Schulze et al. 2008, Stanford & Kenny 2013

+ Bloomfield (1933)'s *Principle of Density* for free
+ Diffusion is straightforward

- Not a lot of control over the network
- Thousands of degrees of freedom

    -> should run many many times

    -> slow

Three Classes of Framework

1. Swarm Frameworks
2. Network Frameworks
    - Speakers are nodes in a graph, edges are possibility of interaction
    - e.g., Baxter et al. 2006, Baxter et al. 2009, Blythe & Croft 2012, Fagyal et al. 2010, Minett & Wang 2008, Kauhanen 2016

Three Classes of Framework

1. Swarm Frameworks
2. Network Frameworks
    - Speakers are nodes in a graph, edges are possibility of interaction
    - e.g., Baxter et al. 2006, Baxter et al. 2009, Blythe & Croft 2012, Fagyal et al. 2010, Minett & Wang 2008, Kauhanen 2016

    \+ Much more control over network structure

    \+ Easy to model concepts from the sociolinguistic lit. (e.g., Milroy & Milroy)

    \- Nodes only interact with immediate neighbours -> slow and less realistic?

    \- Practically implemented as random interactions between neighbours -> same problem as #1

Three Classes of Framework

1. Swarm Frameworks
2. Network Frameworks
3. Algebraic Frameworks
    - Expected outcome of interactions is calculated analytically
    - e.g., Abrams & Stroganz 2003, Baxter et al. 2006, Minett & Wang 2008, Niyogi & Berwick 1997, Yang 2000, Niyogi & Berwick 2009

Three Classes of Framework

1. Swarm Frameworks
2. Network Frameworks
3. Algebraic Frameworks
   - Expected outcome of interactions is calculated analytically
   - e.g., Abrams & Stroganz 2003, Baxter et al. 2006, Minett & Wang 2008, Niyogi & Berwick 1997, Yang 2000, Niyogi & Berwick 2009

+ Closed-form solution rather than simulation -> faster and more direct

- No network structure! Always implemented over perfectly mixed populations

Three Classes of Framework

1. Swarm Frameworks
2. Network Frameworks
3. Algebraic Frameworks

This proliferation of "boutique" frameworks is a problem

- An ad hoc framework risks "overfitting" the pattern
- Comparison between frameworks is challenging

Our Framework

Best of All Worlds

Impose density effects on a network structure and calculate the outcome of each iteration analytically

Best of All Worlds

Impose density effects on a network structure and calculate the outcome of each iteration analytically

Swarm

+ Captures the *Principle of Density*

Network

+ Models key facts about social networks

Algebraic

+ No random process in the core algorithm

The Model

Language change as a two-step loop

1. **Propagation: Variants distribute through the network**
2. **Acquisition: Individuals internalize them**

Vocabulary

L: That which is transmitted

Language ≈ Variant ≈ Sample

G: That which generates/describes/distinguishes L

That which is learned/influenced by L

Grammar ≈ Variety ≈ Latent Variable

Binary G Examples

G: {Merged grammar, Non-merged grammar}

L: Merged or non-merged instances of *cot* and *caught* words

G: {*Dived*-generating grammar, *Dove*-generating grammar}

L: Instances of the past tense of *dive* as *dived* or *dove*

G: {*have*+NEG = *haven't got* grammar, *have*+NEG = *don't have* grammar}

L: Instances of *haven't got* and instances of *don't have*

The Model

Language change as a two-step loop

1. **Propagation**: L distributes through the network
2. **Acquisition**: Individuals react to L to create G

If this were a linear chain,

$$L_0 \rightarrow G_1 \rightarrow L_1 \rightarrow G_2 \rightarrow L_2 \rightarrow \ldots \rightarrow L_n \rightarrow G_{n+1} \rightarrow \ldots$$

The Model

Language change as a two-step loop

1. **Propagation**: L distributes through the network
2. **Acquisition**: Individuals react to L to create G

Generic. Not problem-specific.

Intuition behind Propagation Algorithm

```
For T iterations,
    For the individual at  each node
        Begin travelling;
        While travelling
            Randomly select outgoing edge
                by weight and follow it OR stop;
            Increase chance of stopping next time;
        End
        Interact with the individual at the current
            Node;
    End
End
```

Intuition behind Propagation Algorithm

For T iterations,
    For the individual at each node
        Begin *travelling*;
        While *travelling*
            Randomly select outgoing edge
                by weight and follow it OR stop;
            Increase chance of stopping next time;
        End
        Interact with the individual at the current
            node;
    End
End

Nodes are not individuals.
Individuals “stand on” nodes

Intuition behind Propagation Algorithm

```
For T iterations,
    For the individual at each node
        Begin travelling;
        While travelling
            Randomly select outgoing edge
                by weight and follow it OR stop;
            Increase chance of stopping next time;
        End
        Interact with the individual at the current
            node;
    End
End
```

Individuals “travel” along edges and find someone to interact with

Intuition behind Propagation Algorithm

For T iterations,
For the individual at each node
Begin *travelling*;
While *travelling*
Randomly select outgoing edge
by weight and follow it OR stop;
Increase chance of stopping next time;
End
Interact with the individual at the current
node;
End
End

Individuals connected by shorter or higher weighted paths are more likely to interact.

Intuition behind Propagation Algorithm

For T iterations,

For the individual at each node

[illegible]in *travelling*;

W[illegible] *[illegible]avelling*

[illegible]ct outgoing edge

[illegible]ght and follow it OR stop;

I[illegible]ce of stopping next time;

E[illegible]

[illegible]ract with the in[illegible]l at the current node;

End

End

Rather than simulating interactions in a loop, calculate a closed-form solution

The Propagation Function

$$E = G^T \alpha(I - (1 - \alpha) A)^{-1}$$

The Propagation Function

$$\mathbf{E} = G^T \alpha(I - (1 - \alpha) A)^{-1}$$

The Linguistic Environment

- **E** is a g x n matrix: n individuals, g possible grammars
- For each individual, the proportion of input drawn from each grammar

The Propagation Function

$$E = G^T \alpha(I - (1 - \alpha) A)^{-1}$$

The Linguistic Environment

Distribution of Grammars

- Of the previous generation
- G is an n x g matrix
- Proportions by which each individual produces L

The Propagation Function

$$E = G^T \alpha(I - (1 - \alpha) A)^{-1}$$

The Linguistic Environment

Distribution of Grammars

Interaction Probabilities

- **A is an *n* x *n* adjacency matrix**
- **The probabilities that nodes *i*, *j* interact given that the number of steps travelled declines by a geometric distribution**
- **α parameter from that distribution [0,1]**

The Acquisition Function

- Problem-specific
- Should take E_t as input and produce G_{t+1} as output
- In the simplest case (*neutral change*), $G_{t+1} = E_t^T$
- The following case study uses a *variational learner*

Case Study

Spread of the *Cot-Caught* Merger

Model for Merger Acquisition (Yang 2009)

Learners will acquire the merged grammar iff more than ~17% of their environment is merged

Model for Merger Acquisition (Yang 2009)

Learners will acquire the merged grammar iff more than ~17% of their environment is merged

- Accounts for mergers' tendency to spread (Labov 1994)
- 17% is close to the merged rate estimated in Johnson 2007

Model for Merger Acquisition (Yang 2009)

Learners will acquire the merged grammar iff more than ~17% of their environment is merged

+ Accounts for mergers' tendency to spread (Labov 1994)
+ 17% is close to the merged rate estimated in Johnson 2007
- In a perfectly-mixed model, population will immediately fix at 100% g_+ or g_-

Model for Merger Acquisition (Yang 2009)

Claim: The merged grammar has a processing advantage

Model for Merger Acquisition (Yang 2009)

Claim: The merged grammar has a processing advantage

Claim: Merged listeners have a lower rate of *initial* misinterpretation

Model for Merger Acquisition (Yang 2009)

Claim: The merged grammar has a processing advantage

Claim: Merged listeners have a lower rate of *initial* misinterpretation

Claim: Only *minimal pairs* are relevant

Model for Merger Acquisition (Yang 2009)

Claim: The merged grammar has a processing advantage

Claim: Merged listeners have a lower rate of *initial* misinterpretation

Claim: Only *minimal pairs* are relevant

- If speaker A- and listener B- are both non-merged, B- misunderstands A- at the rate of mishearing one vowel for the other (A- said /ɒ/ but B- heard /ɔ/)

Model for Merger Acquisition (Yang 2009)

Claim: The merged grammar has a processing advantage

Claim: Merged listeners have a lower rate of *initial* misinterpretation

Claim: Only *minimal pairs* are relevant

- If speaker A- and listener B- are both non-merged, B- misunderstands A- at the rate of mishearing one vowel for the other (A- said /ɒ/ but B- heard /ɔ/)
- If A+ speaks to B-, B- initially misunderstands whenever A+ says /ɒ/ when B- expects /ɔ/ and visa-versa

Model for Merger Acquisition (Yang 2009)

Claim: The merged grammar has a processing advantage

Claim: Merged listeners have a lower rate of *initial* misinterpretation

Claim: Only *minimal pairs* are relevant

- If speaker A- and listener B- are both non-merged, B- misunderstands A- at the rate of mishearing one vowel for the other (A- said /ɒ/ but B- heard /ɔ/)
- If A+ speaks to B-, B- initially misunderstands whenever A+ says /ɒ/ when B- expects /ɔ/ and visa-versa
- If A- or A+ speaks to B+, B+ cannot hear A-'s distinctions. Initial misunderstandings come down to lexical access - if the intended meaning is not the most frequent meaning (Carmazza et al 2001)

Variational Model for Merger Acquisition

Probability of initial misunderstanding depends on

- minimal pair frequencies
- mix merged (+) and non-merged (-) speakers in the environment

Variational Model for Merger Acquisition

Probability of initial misunderstanding depends on

- minimal pair frequencies
- mix merged (+) and non-merged (-) speakers in the environment

Using minimal pair frequencies estimated from SUBTLEXus and a variational learner, learners will acquire the merged grammar iff more than ~17% of their environment is merged (Yang 2009)

Acquisition Function

Two Grammars:

Merged grammar g_+

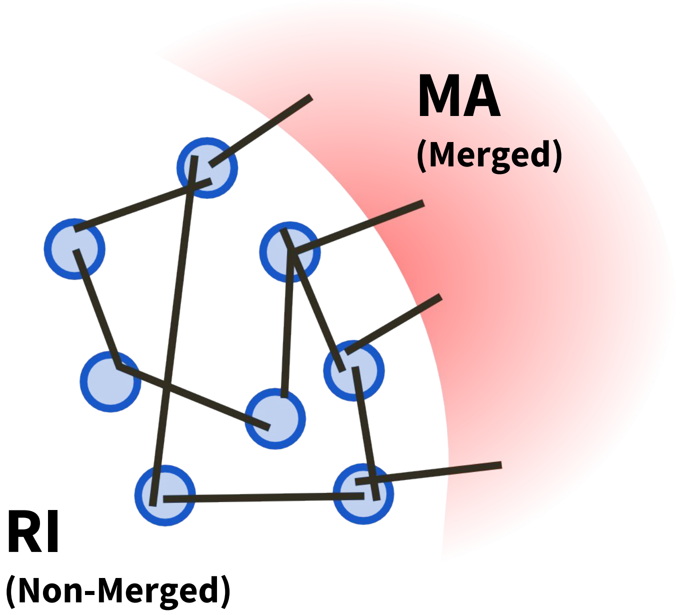
Non-merged grammar g_-

Precomputed Acquisition Function

An individual acquires 100% g_+ if >17% environment is generated by the g_+, else acquire 100% g_-

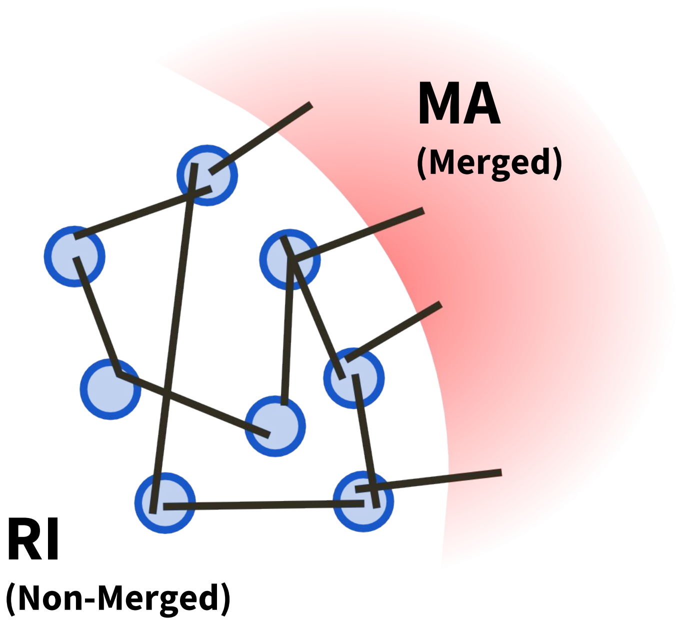
Network Model

- 100 clusters of 75 individuals each
- Each cluster is centralised randomly such that some community members are better connected than others



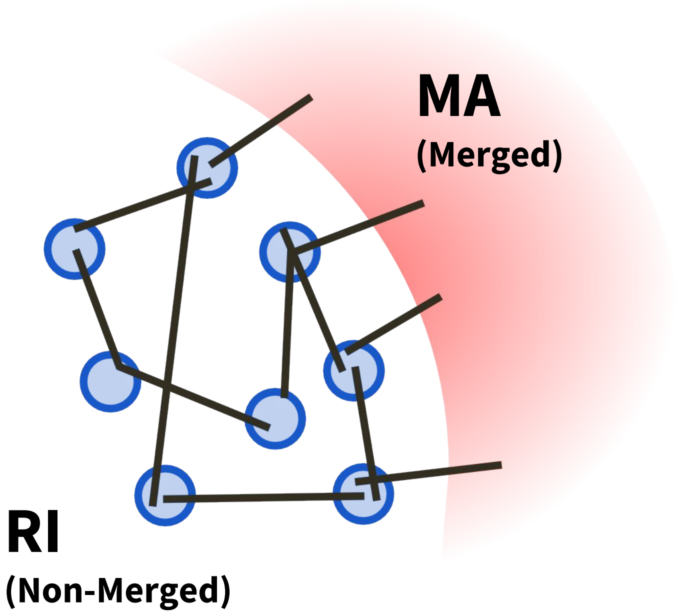
Network Model

- 100 clusters of 75 individuals each
- Each cluster is centralised randomly such that some community members are better connected than others
- One cluster begins 100% merged (Massachusetts)
- The rest start 100% non-merged (Rhode Island)



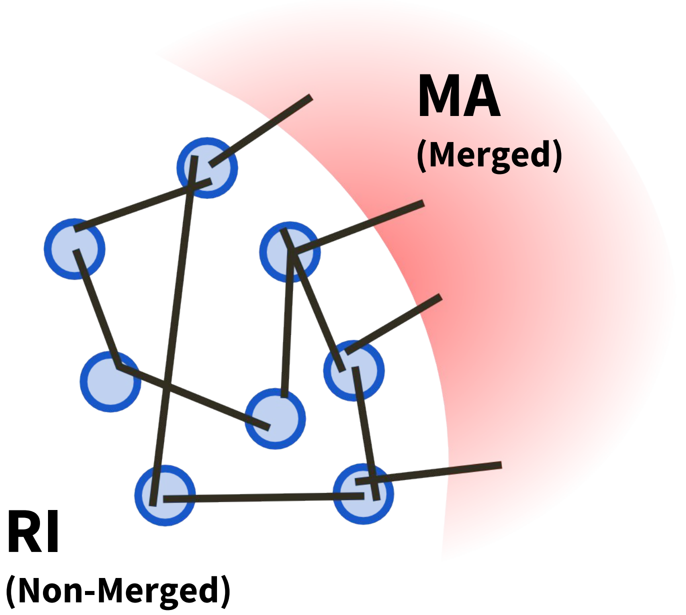
Network Model

- 100 clusters of 75 individuals each
- Each cluster is centralised randomly such that some community members are better connected than others
- One cluster begins 100% merged (Massachusetts)
- The rest start 100% non-merged (Rhode Island)
- Half the RI clusters are connected to the MA cluster (the “Frontier”)



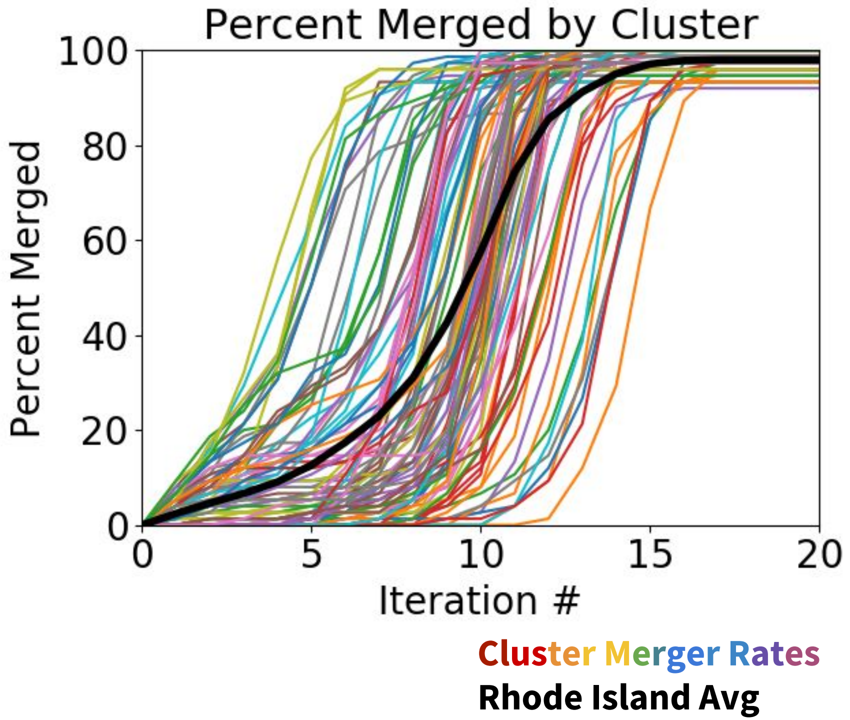
Network Model

- 100 clusters of 75 individuals each
- Each cluster is centralised randomly such that some community members are better connected than others
- One cluster begins 100% merged (Massachusetts)
- The rest start 100% non-merged (Rhode Island)
- Half the RI clusters are connected to the MA cluster (the “Frontier”)
- Two members of each RI cluster are randomly connected to other clusters



Merger Rate in Rhode Island over Time

- The average merger rate across all Rhode Island clusters follows an S-shape
- The 99 RI community cluster curves are also S-shaped
  - Staggered in time
  - Steep slopes = rapid change



Conclusions

The Propagation Function

- Removes the need to simulate interactions
- Is widely applicable rather than made-to-order

The *Cot-Caught* Application

- Predicts behaviour consistent with the empirical data
- And with principles of language change

End

Acknowledgements:

- Charles Yang
- Mitch Marcus
- NDSEG Fellowship (US ARO)

Implementation:

`github.com/jkodner05/NetworksAndLangChange`

Variational Learner (Yang 2000)

- Learners consider multiple grammars g_1, g_2 simultaneously
- $P(g_1) = p$, $P(g_2) = q$, $p+q = 1$

Variational Learner (Yang 2000)

- Learners consider multiple grammars g_1, g_2 simultaneously
- Each g is penalised when it cannot parse an input

- $P(g_1) = p, \quad P(g_2) = q, \quad p+q = 1$
- $p' = \begin{cases} p + \gamma q, & \text{if } g_1 \text{ parses input} \\ (1-\gamma)p, & \text{if } g_1 \text{ fails} \end{cases}$

Variational Learner (Yang 2000)

- Learners consider multiple grammars g_1, g_2 simultaneously
- Each g is penalised when it cannot parse an input
- The g with lower penalty probability has the advantage

- $\mathrm{P}(g_1) = p$, $\mathrm{P}(g_2) = q$, $p+q = 1$
- $p' = \begin{cases} p + \gamma q, & \text{if } g_1 \text{ parses input} \\ (1-\gamma)p, & \text{if } g_1 \text{ fails} \end{cases}$
- $\lim_{t\to\infty} p_t = C_2 / (C_1 + C_2)$
- $\lim_{t\to\infty} q_t = C_1 / (C_2 + C_1)$

Variational Learner (Yang 2000)

- Learners consider multiple grammars g_1, g_2 simultaneously
- Each g is penalised when it cannot parse an input
- The g with lower penalty probability has the advantage
- If mature speakers adopt one grammar categorically, the one with smaller C wins

- $P(g_1) = p, \quad P(g_2) = q, \quad p + q = 1$
- $p' = \begin{cases} p + \gamma q, & \text{if } g_1 \text{ parses input} \\ (1-\gamma)p, & \text{if } g_1 \text{ fails} \end{cases}$
- $\lim_{t\to\infty} p_t = C_2 / (C_1 + C_2)$
- $\lim_{t\to\infty} q_t = C_1 / (C_2 + C_1)$
- $\lim_{t\to\infty} p_t = \begin{cases} 1, & \text{if } C_1 < C_2 \\ 0, & \text{if } C_2 < C_1 \end{cases}$

Variational Model for Merger Acquisition

Penalty probabilities depend on

- minimal pair frequencies
- mix merged (+) and non-merged (-) speakers in the environment

Variational Model for Merger Acquisition

Penalty probabilities depend on

- minimal pair frequencies
- mix merged (+) and non-merged (-) speakers in the environment

m_i, n_i = frequencies of each member of a minimal pair

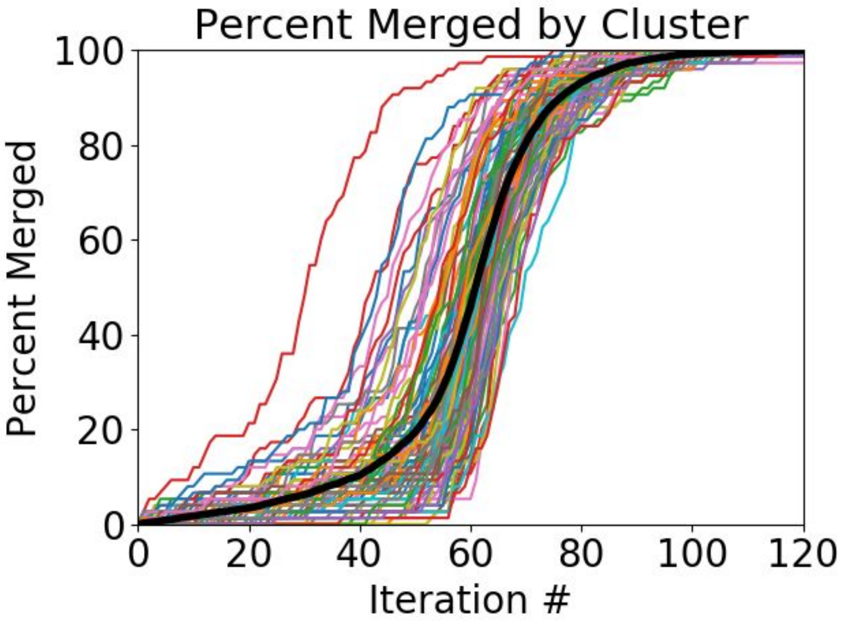
$H = \Sigma_i\, m_i + n_i$

ε = probability of mishearing one vowel for the other

C_+	$= (1/H)\, \Sigma_i \min(m_i, n_i)$	hearing the less freq word
C_-	$= (1/H)\, \Sigma_i [p_+((1-\varepsilon_m)m_i + \varepsilon_n n_i)$	mishearing + input
	$+\, p_-(\varepsilon_m m_i + \varepsilon_n n_i)]$	misinterpreting - input

Results - Updating Connections

- **Social connections change constantly**
- **Rewire the edges (recalculate A) at every iteration**

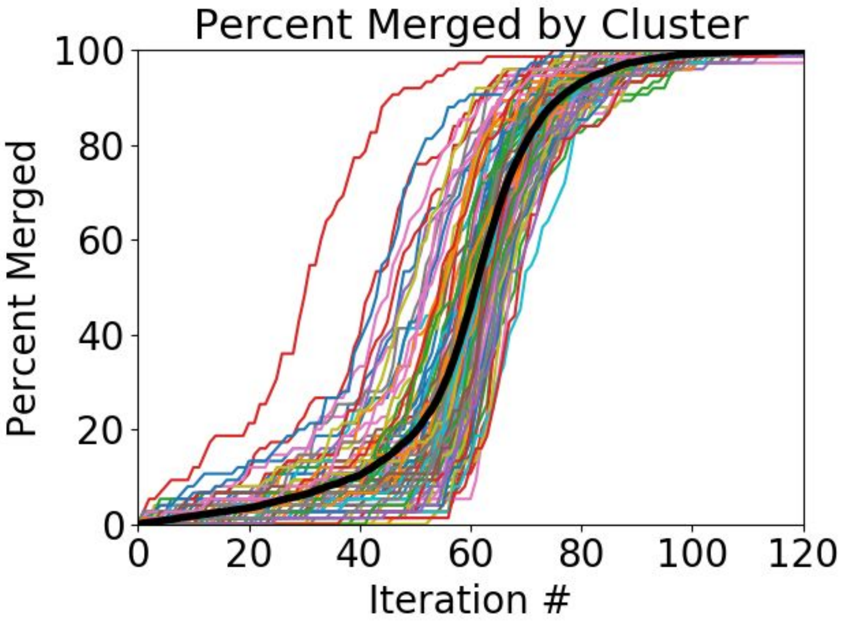


Cluster Merger Rates
Rhode Island Avg

Results - Updating Connections

- Social connections change constantly
- Rewire the edges (recalculate A) at every iteration

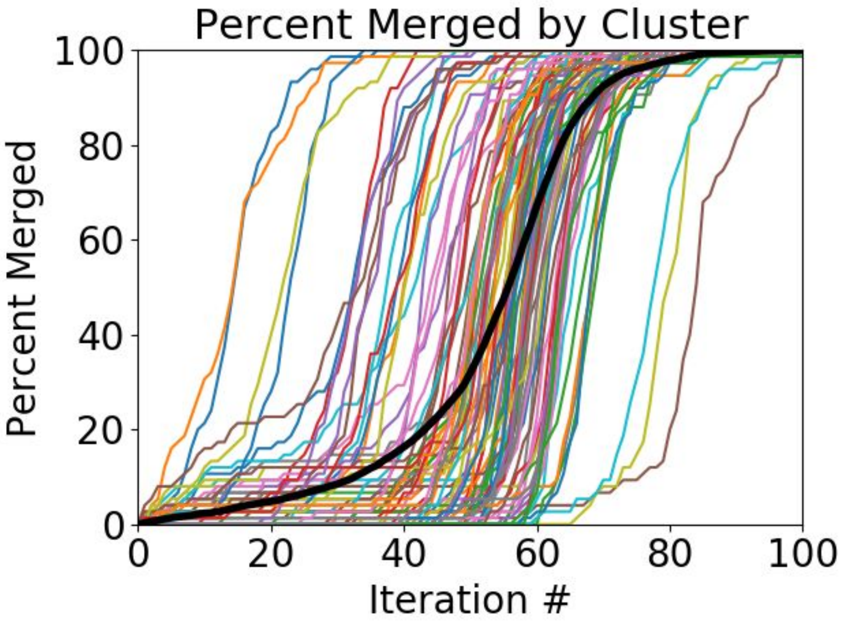
- The outcome is similar, but clusters tipping points are temporally closer
- No cluster remains particularly well or poorly connected for long



Cluster Merger Rates
Rhode Island Avg

Fractional Updating

- The merger spreads rapidly enough to distinguish older and younger siblings
- Only a fraction of the population is of the correct age at any moment
- Update only 10% of random nodes at every iteration

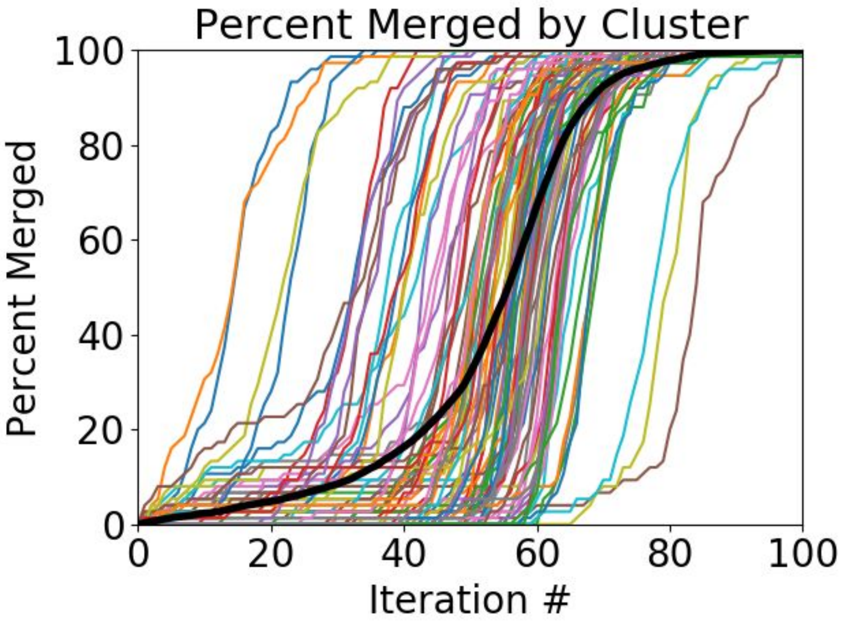


Cluster Merger Rates
Rhode Island Avg

Fractional Updating

- The merger spreads rapidly enough to distinguish older and younger siblings
- Only a fraction of the population is of the correct age at any moment
- Update only 10% of random nodes at every iteration

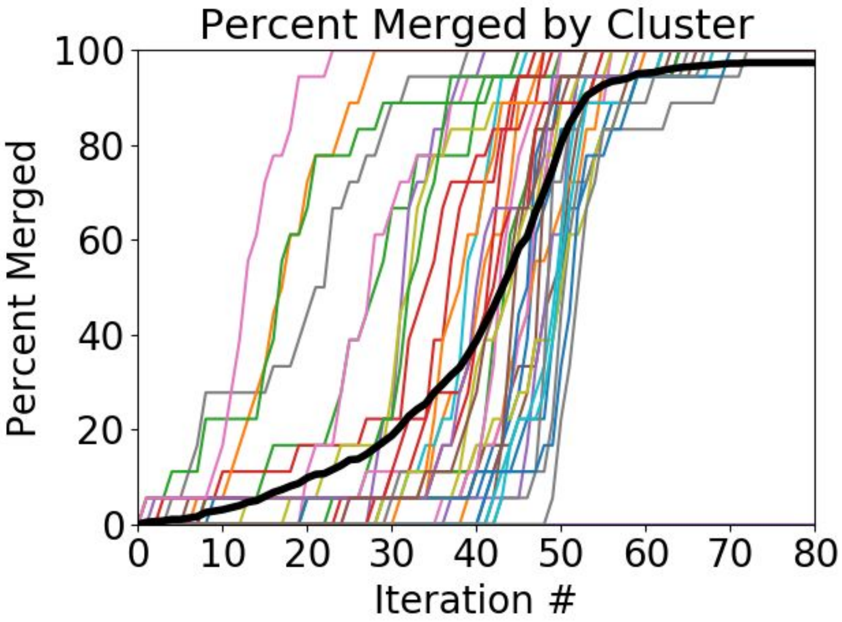
- Similar outcome with wider spread between cluster “tipping points”
- Simulation took about 5x as long because



Cluster Merger Rates
Rhode Island Avg

Results - Network Size

- Tested our network size assumptions
- Repeat the experiment with 40 clusters of 18 individuals each

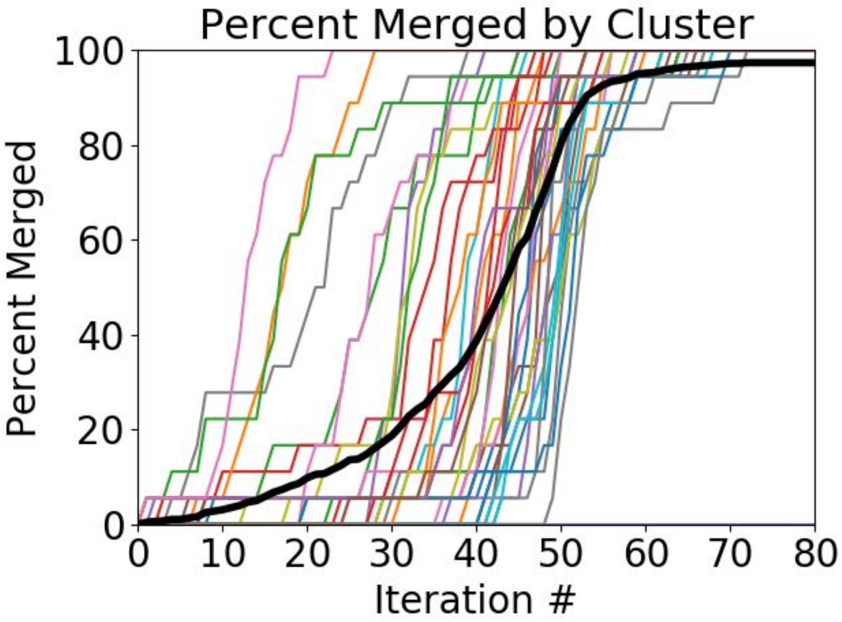


Cluster Merger Rates
Rhode Island Avg

Results - Network Size

- Tested our network size assumptions
- Repeat the experiment with 40 clusters of 18 individuals each

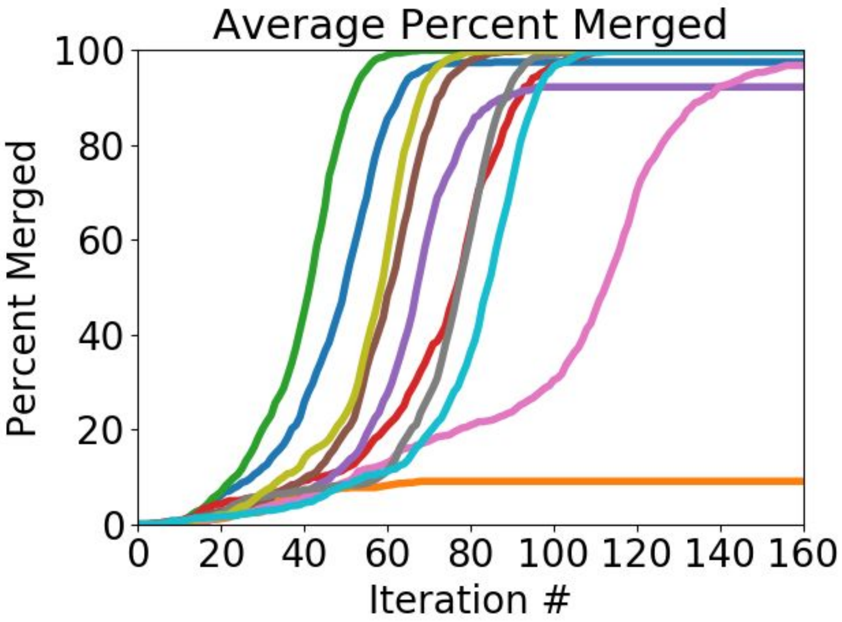
- Qualitatively similar
- The S-shape is less S-shaped
- Individual clusters shows step pattern



Cluster Merger Rates
Rhode Island Avg

Results - Community Averages

- At small network sizes, the community average is more sensitive to random connections
- Repeat the small-scale experiment 10 times

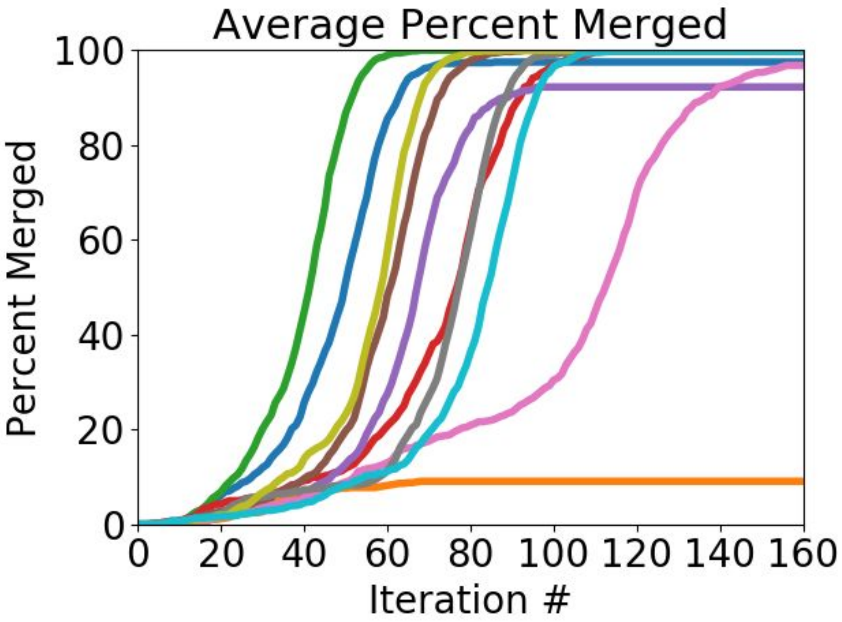


Trial Avgs

Results - Community Averages

- At small network sizes, the community average is more sensitive to random connections
- Repeat the small-scale experiment 10 times

- The slope is ~consistent in most simulations
- A few simulations show aberrant behaviour



Trial Avgs