Computing phonological generalization over real speech exemplars

Robert Kirchner\textsuperscript{1}  Roger K. Moore\textsuperscript{2}

\textsuperscript{1}Linguistics Dept., U. Alberta
\textsuperscript{2}Dept. Computer Science, U. Sheffield

LSA 2009, San Francisco

This research is supported in part by the Social Sciences and Humanities Research Council of Canada, Grant No. 410-2004-1356
Outline

1 Background

2 PEBLS
   - Framing the problem
   - DTW
   - PEBLS: intra-cloud transition matrix
   - Confidence-sensitive alignment

3 Experiment I: simple output generation
   - Questions and method
   - Results and discussion

4 Experiment II: Iterative production

5 Conclusions
Growing interest in Exemplar Theory

- Linguistic categories (e.g. words) are mentally represented as “clouds” of exemplars.
- Potentially affords elegant accounts of
  - frequency effects,
  - sociophonetic variation,
  - incremental sound change.
- Promises a seamless phonetics-phonology interface.
Growing interest in Exemplar Theory

- Linguistic categories (e.g. words) are mentally represented as “clouds” of exemplars.
- Potentially affords elegant accounts of
  - frequency effects,
  - sociophonetic variation,
  - incremental sound change.
- Promises a seamless phonetics-phonology interface.
Growing interest in Exemplar Theory

- Linguistic categories (e.g. words) are mentally represented as “clouds” of exemplars.
- Potentially affords elegant accounts of
  - frequency effects,
  - sociophonetic variation,
  - incremental sound change.
- Promises a seamless phonetics-phonology interface.
Growing interest in Exemplar Theory

- Linguistic categories (e.g. words) are mentally represented as “clouds” of exemplars.
- Potentially affords elegant accounts of
  - frequency effects,
  - sociophonetic variation,
  - incremental sound change.
- Promises a seamless phonetics-phonology interface.
Growing interest in Exemplar Theory

- Linguistic categories (e.g. words) are mentally represented as “clouds” of exemplars.
- Potentially affords elegant accounts of
  - frequency effects,
  - sociophonetic variation,
  - incremental sound change.
- Promises a seamless phonetics-phonology interface.
Growing interest in Exemplar Theory

- Linguistic categories (e.g., words) are mentally represented as “clouds” of exemplars.
- Potentially affords elegant accounts of
  - frequency effects,
  - sociophonetic variation,
  - incremental sound change.
- Promises a seamless phonetics-phonology interface.
Growing interest in Exemplar Theory

- Linguistic categories (e.g. words) are mentally represented as “clouds” of exemplars.

- Potentially affords elegant accounts of
  - frequency effects,
  - sociophonetic variation,
  - incremental sound change.

- Promises a seamless phonetics-phonology interface.


Growing interest in Exemplar Theory

- Linguistic categories (e.g. words) are mentally represented as “clouds” of exemplars.
- Potentially affords elegant accounts of
  - frequency effects,
  - sociophonetic variation,
  - incremental sound change.
- Promises a seamless phonetics-phonology interface.
But no explicit production model

- An adequate exemplar-based production model must be able to *generalize*:
  - i.e. to find patterns over exemplars and enforce them on outputs,
  - otherwise the system is strictly limited to past experiences.
- Pierrehumbert 2001: without generalization, categories iteratively increase their variances, leading to catastrophic neutralization.

Pierrehumbert therefore proposes *averaging* over a group of exemplars.

- applied to static data;
- not clear how to extend to speech signals: variable-length time-series data.

- Without an explicit production model for real speech, Exemplar Theory’s claims cannot be rigourously tested.
But no explicit production model

- An adequate exemplar-based production model must be able to generalize:
  - i.e. to find patterns over exemplars and enforce them on outputs,
  - otherwise the system is strictly limited to past experiences.
  - Pierrehumbert 2001: without generalization, categories iteratively increase their variances, leading to catastrophic neutralization.

- Pierrehumbert therefore proposes averaging over a group of exemplars.
  - applied to static data;
  - not clear how to extend to speech signals: variable-length time-series data.

- Without an explicit production model for real speech, Exemplar Theory’s claims cannot be rigourously tested.
But no explicit production model

- An adequate exemplar-based production model must be able to generalize:
  - i.e. to find patterns over exemplars and enforce them on outputs,
  - otherwise the system is strictly limited to past experiences.
  - Pierrehumbert 2001: without generalization, categories iteratively increase their variances, leading to catastrophic neutralization.
  - Pierrehumbert therefore proposes averaging over a group of exemplars.
    - applied to static data;
    - not clear how to extend to speech signals: variable-length time-series data.

- Without an explicit production model for real speech, Exemplar Theory’s claims cannot be rigorously tested.
But no explicit production model

- An adequate exemplar-based production model must be able to generalize:
  - i.e. to find patterns over exemplars and enforce them on outputs,
  - otherwise the system is strictly limited to past experiences.
  - Pierrehumbert 2001: without generalization, categories iteratively increase their variances, leading to catastrophic neutralization.

- Pierrehumbert therefore proposes averaging over a group of exemplars.
  - applied to static data;
  - not clear how to extend to speech signals: variable-length time-series data.

- Without an explicit production model for real speech, Exemplar Theory’s claims cannot be rigourously tested.
An adequate exemplar-based production model must be able to **generalize**:
- i.e. to find patterns over exemplars and enforce them on outputs,
- otherwise the system is strictly limited to past experiences.
- Pierrehumbert 2001: without generalization, categories iteratively increase their variances, leading to catastrophic neutralization.

Pierrehumbert therefore proposes **averaging** over a group of exemplars.
- applied to static data;
- not clear how to extend to speech signals: variable-length time-series data.

Without an explicit production model for real speech, Exemplar Theory’s claims cannot be rigorously tested.
But no explicit production model

- An adequate exemplar-based production model must be able to *generalize*:
  - i.e. to find patterns over exemplars and enforce them on outputs,
  - otherwise the system is strictly limited to past experiences.
- Pierrehumbert 2001: without generalization, categories iteratively increase their variances, leading to catastrophic neutralization.

- Pierrehumbert therefore proposes *averaging* over a group of exemplars.
  - applied to static data;
  - not clear how to extend to speech signals: variable-length time-series data.

- Without an explicit production model for real speech, Exemplar Theory’s claims cannot be rigorously tested.
An adequate exemplar-based production model must be able to generalize:
- i.e. to find patterns over exemplars and enforce them on outputs,
- otherwise the system is strictly limited to past experiences.
- Pierrehumbert 2001: without generalization, categories iteratively increase their variances, leading to catastrophic neutralization.

Pierrehumbert therefore proposes averaging over a group of exemplars.
- applied to static data;
- not clear how to extend to speech signals: variable-length time-series data.

Without an explicit production model for real speech, Exemplar Theory's claims cannot be rigourously tested.
But no explicit production model

- An adequate exemplar-based production model must be able to generalize:
  - i.e. to find patterns over exemplars and enforce them on outputs,
  - otherwise the system is strictly limited to past experiences.
- Pierrehumbert 2001: without generalization, categories iteratively increase their variances, leading to catastrophic neutralization.
- Pierrehumbert therefore proposes averaging over a group of exemplars.
  - applied to static data;
  - not clear how to extend to speech signals: variable-length time-series data.
- Without an explicit production model for real speech, Exemplar Theory’s claims cannot be rigourously tested.
Temptation: appeal to phonological units

- Proposed (but not computationally fleshed out) by Pierrehumbert (2002):
  - trying to avoid the time dimension,
  - by chunking the signal into quasi-static portions, i.e. phone units,
  - which can then be treated as static data points, as in Pierrehumbert 2001.

- But this seems contrary to the spirit of Exemplar Theory:
  - phonological units should emerge bottom-up from comparison over the exemplars.

- Fails to do justice to the rich dynamic structure of speech.
- We resist this temptation, looking instead for an inherently dynamic model.
Temptation: appeal to phonological units

- Proposed (but not computationally fleshed out) by Pierrehumbert (2002):
  - trying to avoid the time dimension,
  - by chunking the signal into quasi-static portions, i.e. phone units,
  - which can then be treated as static data points, as in Pierrehumbert 2001.

- But this seems contrary to the spirit of Exemplar Theory:
  - phonological units should emerge bottom-up from comparison over the exemplars.

- Fails to do justice to the rich dynamic structure of speech.
- We resist this temptation, looking instead for an inherently dynamic model.
Temptation: appeal to phonological units

- Proposed (but not computationally fleshed out) by Pierrehumbert (2002):
  - trying to avoid the time dimension,
  - by chunking the signal into quasi-static portions, i.e. phone units,
  - which can then be treated as static data points, as in Pierrehumbert 2001.

- But this seems contrary to the spirit of Exemplar Theory:
  - phonological units should emerge bottom-up from comparison over the exemplars.

- Fails to do justice to the rich dynamic structure of speech.

- We resist this temptation, looking instead for an inherently dynamic model.
Temptation: appeal to phonological units

- Proposed (but not computationally fleshed out) by Pierrehumbert (2002):
  - trying to avoid the time dimension,
  - by chunking the signal into quasi-static portions, i.e. phone units,
  - which can then be treated as static data points, as in Pierrehumbert 2001.

- But this seems contrary to the spirit of Exemplar Theory:
  - phonological units should emerge bottom-up from comparison over the exemplars.

- Fails to do justice to the rich dynamic structure of speech.

- We resist this temptation, looking instead for an inherently dynamic model.
Temptation: appeal to phonological units

- Proposed (but not computationally fleshed out) by Pierrehumbert (2002):
  - trying to avoid the time dimension,
  - by chunking the signal into quasi-static portions, i.e. phone units,
  - which can then be treated as static data points, as in Pierrehumbert 2001.

- But this seems contrary to the spirit of Exemplar Theory:
  - phonological units should emerge bottom-up from comparison over the exemplars.
  - Fails to do justice to the rich dynamic structure of speech.
  - We resist this temptation, looking instead for an inherently dynamic model.
Temptation: appeal to phonological units

- Proposed (but not computationally fleshed out) by Pierrehumbert (2002):
  - trying to avoid the time dimension,
  - by chunking the signal into quasi-static portions, i.e. phone units,
  - which can then be treated as static data points, as in Pierrehumbert 2001.

- But this seems contrary to the spirit of Exemplar Theory:
  - phonological units should emerge bottom-up from comparison over the exemplars.

- Fails to do justice to the rich dynamic structure of speech.
- We resist this temptation, looking instead for an inherently dynamic model.
Temptation: appeal to phonological units

- Proposed (but not computationally fleshed out) by Pierrehumbert (2002):
  - trying to avoid the time dimension,
  - by chunking the signal into quasi-static portions, i.e. phone units,
  - which can then be treated as static data points, as in Pierrehumbert 2001.

- But this seems contrary to the spirit of Exemplar Theory:
  - phonological units should emerge bottom-up from comparison over the exemplars.

- Fails to do justice to the rich dynamic structure of speech.
  - We resist this temptation, looking instead for an inherently dynamic model.
Temptation: appeal to phonological units

• Proposed (but not computationally fleshed out) by Pierrehumbert (2002):
  • trying to avoid the time dimension,
  • by chunking the signal into quasi-static portions, i.e. phone units,
  • which can then be treated as static data points, as in Pierrehumbert 2001.

• But this seems contrary to the spirit of Exemplar Theory:
  • phonological units should emerge bottom-up from comparison over the exemplars.

• Fails to do justice to the rich dynamic structure of speech.

• We resist this temptation, looking instead for an inherently dynamic model.
Exemplars in memory are already assigned word category labels by the recognition model (not discussed).

Randomly select an exemplar from the target word class as the input.

The remainder of the exemplars are the cloud.

- In today’s talk, the cloud is restricted to other exemplars of the same word category.
- The cloud thus contain exemplars which are similar, but never identical, to the input.
Exemplars in memory are already assigned word category labels by the recognition model (not discussed).

Randomly select an exemplar from the target word class as the input.

The remainder of the exemplars are the cloud.

- In today’s talk, the cloud is restricted to other exemplars of the same word category.
- The cloud thus contain exemplars which are similar, but never identical, to the input.
Exemplars in memory are already assigned word category labels by the recognition model (not discussed).

Randomly select an exemplar from the target word class as the input.

The remainder of the exemplars are the cloud.

In today’s talk, the cloud is restricted to other exemplars of the same word category.

The cloud thus contain exemplars which are similar, but never identical, to the input.
Exemplars in memory are already assigned word category labels by the recognition model (not discussed).

Randomly select an exemplar from the target word class as the input.

The remainder of the exemplars are the cloud.

* In today’s talk, the cloud is restricted to other exemplars of the same word category.
* The cloud thus contain exemplars which are similar, but never identical, to the input.
Exemplars in memory are already assigned word category labels by the recognition model (not discussed).

Randomly select an exemplar from the target word class as the *input*.

The remainder of the exemplars are the *cloud*.

- In today’s talk, the cloud is restricted to other exemplars of the same word category.
- The cloud thus contain exemplars which are similar, but never identical, to the input.
Background
PEBLS
Experiment I: simple output generation
Experiment II: Iterative production
Conclusions
Framing the problem
DTW
PEBLS: intra-cloud transition matrix
Confidence-sensitive alignment

Output as an alignment schematically illustrated

Output generation: finding optimal alignment of input with cloud.

Numbers indicate corresponding subsequences within the input and cloud, and the concatenation of these subsequences to form the output. Letters show the particular exemplar from which each subsequence was taken.
Outline

1. Background

2. PEBLS
   - Framing the problem
   - DTW
   - PEBLS: intra-cloud transition matrix
   - Confidence-sensitive alignment

3. Experiment I: simple output generation
   - Questions and method
   - Results and discussion

4. Experiment II: Iterative production

5. Conclusions
Introducing dynamic time warping (DTW)

- A computational technique for optimally aligning two variable-length signals A and B,
  - locally stretching or shrinking subsequences within A to best fit B, or vice-versa.
- Presupposes some meaningful measure of distance between timepoints of each of the signals.
  - e.g. let A and B be spectrograms,
  - take Euclidean distance of every frame of A from every frame of B to construct a distance matrix.
Introducing dynamic time warping (DTW)

- A computational technique for optimally aligning two variable-length signals A and B,
  - locally stretching or shrinking subsequences within A to best fit B, or vice-versa.
- Presupposes some meaningful measure of distance between timepoints of each of the signals.
  - e.g. let A and B be spectrograms,
  - take Euclidean distance of every frame of A from every frame of B to construct a distance matrix.

Robert Kirchner, Roger K. Moore
Computing over exemplars
Introducing dynamic time warping (DTW)

- A computational technique for optimally aligning two variable-length signals A and B,
  - locally stretching or shrinking subsequences within A to best fit B, or vice-versa.
- Presupposes some meaningful measure of distance between timepoints of each of the signals.
  - e.g. let A and B be spectrograms,
  - take Euclidean distance of every frame of A from every frame of B to construct a distance matrix.
Introducing dynamic time warping (DTW)

- A computational technique for optimally aligning two variable-length signals A and B,
  - locally stretching or shrinking subsequences within A to best fit B, or vice-versa.
- Presupposes some meaningful measure of distance between timepoints of each of the signals.
  - e.g. let A and B be spectrograms,
    - take Euclidean distance of every frame of A from every frame of B to construct a distance matrix.
Introducing dynamic time warping (DTW)

- A computational technique for optimally aligning two variable-length signals A and B,
  - locally stretching or shrinking subsequences within A to best fit B, or vice-versa.

- Presupposes some meaningful measure of distance between timepoints of each of the signals.
  - e.g. let A and B be spectrograms,
  - take Euclidean distance of every frame of A from every frame of B to construct a *distance matrix*. 
DTW: How did I get here?

DTW breaks a complex problem down into possible sub-solutions, and for each sub-solution,

- asks “how did I (optimally) get here?”
- and records the results.

Each sub-solution corresponds to a cell in the distance matrix, which can be reached from at most three other cells:
DTW: How did I get here?

- DTW breaks a complex problem down into possible sub-solutions, and for each sub-solution,
  - asks “how did I (optimally) get here?”
  - and records the results.
- Each sub-solution corresponds to a cell in the distance matrix, which can be reached from at most three other cells:
**DTW: How did I get here?**

- DTW breaks a complex problem down into possible sub-solutions, and for each sub-solution,
  - asks “how did I (optimally) get here?”
  - and records the results.
- Each sub-solution corresponds to a cell in the distance matrix, which can be reached from at most three other cells:
DTW: How did I get here?

- DTW breaks a complex problem down into possible sub-solutions, and for each sub-solution,
  - asks “how did I (optimally) get here?”
  - and records the results.
- Each sub-solution corresponds to a cell in the distance matrix, which can be reached from at most three other cells:

![Matrix Diagram]

```
<table>
<thead>
<tr>
<th></th>
<th>i-1</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>j-1</td>
<td>3.07 substitution</td>
<td>8.41 deletion</td>
</tr>
<tr>
<td></td>
<td>10.79 insertion</td>
<td>4.37</td>
</tr>
</tbody>
</table>
```
DTW: cumulative distance, decision

- The cumulative distance $D_{i,j} = \min(D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}) + d_{i,j}$, where $d$ denotes raw distance.
  - $D_{i-1,j-1}$ “cost of getting there”
  - $d_{i,j}$ “cost of being there”

- We also record the decision: which cell has the minimum cumulative distance.
The **cumulative** distance $D_{i,j} = \min(D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}) + d_{i,j}$, where $d$ denotes raw distance.

- $D_{i-1,j-1}$ "cost of getting there"
- $d_{i,j}$ "cost of being there"

We also record the *decision*: which cell has the minimum cumulative distance.
DTW: cumulative distance, decision

- The cumulative distance $D_{i,j} = \min(D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}) + d_{i,j}$, where $d$ denotes raw distance.
  - $D_{i-1,j-1}$ “cost of getting there”
  - $d_{i,j}$ “cost of being there”

- We also record the decision: which cell has the minimum cumulative distance.
DTW: cumulative distance, decision

- The *cumulative distance* \( D_{i,j} = \min(D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}) + d_{i,j} \), where \( d \) denotes raw distance.
  - \( D_{i-1,j-1} \) “cost of getting there”
  - \( d_{i,j} \) “cost of being there”
- We also record the *decision*: which cell has the minimum cumulative distance.
DTW: traceback

- The algorithm proceeds iteratively from upper left to lower right.
- Once all the cumulative distances have been computed, starting at the bottom-right cell, iteratively trace back the chain of decisions that led there.
- This iterative traceback gives us the alignment.
DTW: traceback

- The algorithm proceeds iteratively from upper left to lower right.
- Once all the cumulative distances have been computed, starting at the bottom-right cell, iteratively trace back the chain of decisions that led there.
- This iterative traceback gives us the alignment.
The algorithm proceeds iteratively from upper left to lower right.

Once all the cumulative distances have been computed, starting at the bottom-right cell, iteratively trace back the chain of decisions that led there.

This iterative traceback gives us the alignment.
Outline

1. Background
2. PEBLS
   - Framing the problem
   - DTW
   - PEBLS: intra-cloud transition matrix
   - Confidence-sensitive alignment
3. Experiment I: simple output generation
   - Questions and method
   - Results and discussion
4. Experiment II: Iterative production
5. Conclusions
Permissible transitions

- DTW aligns whole signal to whole signal.
- PEBLS, however, must be able to align matching subsequences, even with temporal reversals.
  - Hence, all transitions are possible,
  - but some transitions are more permissible than others,
    - namely, transitions which are similar to those in actual exemplars in the cloud.
  - Similarity of frames $i,j$ is related to distance as $s_{i,j} = \exp(-cd_{i,j})$, where $c$ scales steepness of drop-off.

Robert Kirchner, Roger K. Moore
Computing over exemplars
Permissible transitions

- DTW aligns whole signal to whole signal.
- PEBLS, however, must be able to align matching subsequences, even with temporal reversals.
  - Hence, all transitions are possible,
  - but some transitions are more permissible than others,
    - namely, transitions which are similar to those in actual exemplars in the cloud.
    - Similarity of frames $i,j$ is related to distance as $s_{i,j} = \exp(-cd_{i,j})$, where $c$ scales steepness of drop-off.
Permissible transitions

- DTW aligns whole signal to whole signal.
- PEBLS, however, must be able to align matching subsequences, even with temporal reversals.
  - Hence, all transitions are possible,
  - but some transitions are more permissible than others,
    - namely, transitions which are similar to those in actual exemplars in the cloud.
    - Similarity of frames $i, j$ is related to distance as $s_{i,j} = \exp(-cd_{i,j})$, where $c$ scales steepness of drop-off.
Permissible transitions

- DTW aligns whole signal to whole signal.
- PEBLS, however, must be able to align matching subsequences, even with temporal reversals.
  - Hence, all transitions are possible,
  - but some transitions are more permissible than others,
    - namely, transitions which are similar to those in actual exemplars in the cloud.
    - Similarity of frames $i,j$ is related to distance as $s_{i,j} = \exp(-cd_{i,j})$, where $c$ scales steepness of drop-off.
Permissible transitions

- DTW aligns whole signal to whole signal.
- PEBLS, however, must be able to align matching subsequences, even with temporal reversals.
  - Hence, all transitions are possible,
  - but some transitions are more permissible than others,
    - namely, transitions which are similar to those in actual exemplars in the cloud.
    - Similarity of frames $i,j$ is related to distance as $s_{i,j} = \exp(-cd_{i,j})$, where $c$ scales steepness of drop-off.
Permissible transitions

- DTW aligns whole signal to whole signal.
- PEBLS, however, must be able to align matching subsequences, even with temporal reversals.
  - Hence, all transitions are possible,
  - but some transitions are more permissible than others,
    - namely, transitions which are similar to those in actual exemplars in the cloud.
    - Similarity of frames $i,j$ is related to distance as $s_{i,j} = \exp(-cd_{i,j})$, where $c$ scales steepness of drop-off.
Transition network

- To compute this permissibility, we construct an intra-cloud transition network: a similarity matrix of the entire cloud to itself, offset by one frame.
- Cell \((i,j)\) of this matrix encodes the similarity of \(i\) to the frame that immediately precedes \(j\).
- Encodes not only how the input aligns with each exemplar in the cloud,
  - but also how the cloud aligns with itself,
- Getting emergent structure from self-similarity within the data.
To compute this permissibility, we construct an intra-cloud transition network: a similarity matrix of the entire cloud to itself, offset by one frame.

Cell \((i,j)\) of this matrix encodes the *similarity of \(i\) to the frame that immediately precedes \(j\).*

encodes not only how the input aligns with each exemplar in the cloud,

but also how the cloud aligns with itself,

gaining emergent structure from self-similarity within the data.
Transition network

- To compute this permissibility, we construct an intra-cloud transition network: a similarity matrix of the entire cloud to itself, offset by one frame.
- Cell \((i,j)\) of this matrix encodes the *similarity of i to the frame that immediately precedes j*.
- encodes not only how the input aligns with each exemplar in the cloud,
  - but also how the cloud aligns with itself,
  - getting emergent structure from self-similarity within the data.
Transition network

- To compute this permissibility, we construct an intra-cloud transition network: a similarity matrix of the entire cloud to itself, offset by one frame.
- Cell \((i,j)\) of this matrix encodes the similarity of \(i\) to the frame that immediately precedes \(j\).
- Encodes not only how the input aligns with each exemplar in the cloud,
  - but also how the cloud aligns with itself,
  - getting emergent structure from self-similarity within the data.
Transition network

- To compute this permissibility, we construct an intra-cloud transition network: a similarity matrix of the entire cloud to itself, offset by one frame.
- Cell \((i,j)\) of this matrix encodes the similarity of \(i\) to the frame that immediately precedes \(j\).
- Encodes not only how the input aligns with each exemplar in the cloud,
  - but also how the cloud aligns with itself,
  - getting emergent structure from self-similarity within the data.
Cumulative similarity in PEBLS

- Cumulative similarity $S$ of the $v^{th}$ frame of the input to the $u^{th}$ frame of the cloud can be calculated as

$$S_{u,v} = \max_{i=1}^{U} (s_{i,v-1} t_{i,u}) + s_{u,v}$$

where $U$ is the number of frames in the whole cloud.

- The decision is given by $\arg\max_{i=1}^{U} (s_{i,v-1} t_{i,u})$
Cumulative similarity in PEBLS

- Cumulative similarity $S$ of the $\nu^{th}$ frame of the input to the $u^{th}$ frame of the cloud can be calculated as

$$S_{u,\nu} = \max_{i=1}^{U} (s_{i,\nu-1}t_{i,u}) + s_{u,\nu}$$

where $U$ is the number of frames in the whole cloud.

- The decision is given by $\arg \max_{i=1}^{U} (s_{i,\nu-1}t_{i,u})$
What’s wrong with *max*?

- The model presented thus far finds the maximum similarity alignment between input and intra-cloud transition network.
- What we want, though, is an alignment that reflects frame sequences which are *typical* of the cloud,
  - analogous to the statistical notion of *confidence* that a particular sample reflects the distribution of an underlying population.
What’s wrong with \textit{max}?

- The model presented thus far finds the maximum similarity alignment between input and intra-cloud transition network.
- What we want, though, is an alignment that reflects frame sequences which are \textit{typical} of the cloud,
  - analogous to the statistical notion of \textit{confidence} that a particular sample reflects the distribution of an underlying population.
What’s wrong with $max$?

- The model presented thus far finds the maximum similarity alignment between input and intra-cloud transition network.
- What we want, though, is an alignment that reflects frame sequences which are *typical* of the cloud,
  - analogous to the statistical notion of *confidence* that a particular sample reflects the distribution of an underlying population.
Confidence-sensitive alignment

- Confidence score obtained by *hierarchically clustering* getting-there scores from previous frame.
  - Optimal cluster \( w = \arg \max_i \left( \frac{\mu_i N_i}{\sigma_i^2 + 1} \right) \)
    - where \( \mu_i \) is the mean getting-there score, \( N_i \) the size, and \( \sigma_i^2 \) the variance, of cluster \( i \).
  - The optimal getting-there score is then \( \mu_w \), and the decision is \( \arg \min_U \left( \sum_{i=1}^U |u_i - \mu_w| \right) \).
  - Allows trade-offs between similarity and density (size over variance).
Confidence-sensitive alignment

- Confidence score obtained by *hierarchically clustering* getting-there scores from previous frame.
  
  - Optimal cluster $w = \arg \max_{i=1}^{2U-1} \left( \frac{\mu_i N_i}{\sigma_i^2 + 1} \right)$

  where $\mu_i$ is the mean getting-there score, $N_i$ the size, and $\sigma_i^2$ the variance, of cluster $i$.

- The optimal getting-there score is then $\mu_w$, and the decision is $\arg \min_{i=1}^{U} (|u_i - \mu_w|)$.

- Allows trade-offs between similarity and density (size over variance).
Confidence-sensitive alignment

- Confidence score obtained by *hierarchically clustering* getting-there scores from previous frame.
  - Optimal cluster \( w = \arg \max_{i=1}^{2U-1} \left( \frac{\mu_i N_i}{\sigma_i^2 + 1} \right) \)
    
    where \( \mu_i \) is the mean getting-there score, \( N_i \) the size, and \( \sigma_i^2 \) the variance, of cluster \( i \).

- The optimal getting-there score is then \( \mu_w \), and the decision is 
  
  \[
  U \arg \min_{i=1}^{U} (|u_i - \mu_w|).
  \]

- Allows trade-offs between similarity and density (size over variance).
Outline

1. Background
2. PEBLS
   - Framing the problem
   - DTW
   - PEBLS: intra-cloud transition matrix
   - Confidence-sensitive alignment
3. Experiment I: simple output generation
   - Questions and method
   - Results and discussion
4. Experiment II: Iterative production
5. Conclusions

Robert Kirchner, Roger K. Moore  Computing over exemplars
Questions

1. As a threshold matter, can PEBLS generate appropriate outputs for given target words, which can be resynthesized into reasonably natural sounding speech?

2. Do PEBLS’ outputs show generalization?
   - Focussing on a pattern of allophonic spirantization of /k/ in intervocalic position.
Questions

1. As a threshold matter, can PEBLS generate appropriate outputs for given target words, which can be resynthesized into reasonably natural sounding speech?

2. Do PEBLS’ outputs show generalization?
   - focussing on a pattern of allophonic spirantization of /k/ in intervocalic position.
Questions

1. As a threshold matter, can PEBLS generate appropriate outputs for given target words, which can be resynthesized into reasonably natural sounding speech?

2. Do PEBLS’ outputs show generalization?
   - Focussing on a pattern of allophonic spirantization of /k/ in intervocalic position.
## Wordlist

<table>
<thead>
<tr>
<th>Pattern-conforming</th>
<th>Pattern-violating</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Intervocalic [x]</strong></td>
<td><strong>Intervocalic [k]</strong></td>
</tr>
<tr>
<td>æxæ</td>
<td>æks</td>
</tr>
<tr>
<td>æxe</td>
<td>ækt</td>
</tr>
<tr>
<td>æxi</td>
<td>eks</td>
</tr>
<tr>
<td>exæ</td>
<td>ekt</td>
</tr>
<tr>
<td>exe</td>
<td>iks</td>
</tr>
<tr>
<td>exi</td>
<td>ikt</td>
</tr>
<tr>
<td>ixi</td>
<td>skæ</td>
</tr>
<tr>
<td>ixæ</td>
<td>ske</td>
</tr>
<tr>
<td>ixe</td>
<td>ski</td>
</tr>
</tbody>
</table>

Robert Kirchner, Roger K. Moore
Computing over exemplars
Eighteen clouds were constructed, consisting of

- all ten tokens of each of the pattern-conforming words, plus
- one token each of the pattern-violating words.

Each cloud thus reflects a strong, albeit variable pattern of allophonic intervocalic spirantization.
Eighteen clouds were constructed, consisting of

- all ten tokens of each of the pattern-conforming words, plus
- one token each of the pattern-violating words.

Each cloud thus reflects a strong, albeit variable pattern of allophonic intervocalic spirantization.
Eighteen clouds were constructed, consisting of:

- all ten tokens of each of the pattern-conforming words, plus
- one token each of the pattern-violating words.

Each cloud thus reflects a strong, albeit variable pattern of allophonic intervocalic spirantization.
Eighteen clouds were constructed, consisting of
- all ten tokens of each of the pattern-conforming words, plus
- one token each of the pattern-violating words.

Each cloud thus reflects a strong, albeit variable pattern of allophonic intervocalic spirantization.
What does *generalization* mean?

- We operationalize the notion of generalization as follows:
  - If an input is selected which violates the spirantization pattern,
  - and the resulting PEBLS output conforms to the pattern,
  - then PEBLS has generalized the pattern.
What does *generalization* mean?

- We operationalize the notion of generalization as follows:
  - If an input is selected which violates the spirantization pattern,
  - and the resulting PEBLS output conforms to the pattern,
  - then PEBLS has generalized the pattern.
What does *generalization* mean?

- We operationalize the notion of generalization as follows:
  - If an input is selected which violates the spirantization pattern,
  - and the resulting PEBLS output conforms to the pattern,
  - then PEBLS has generalized the pattern.
What does *generalization* mean?

- We operationalize the notion of generalization as follows:
  - If an input is selected which violates the spirantization pattern,
  - and the resulting PEBLS output conforms to the pattern,
  - then PEBLS has generalized the pattern.
Design details

- The audio signals were preprocessed into frames of 13 mel-frequency cepstral coefficients (MFCCs).
- The similarity drop-off parameter $c$ was set to 30.
- In addition, a similarity threshold of 0.1 was imposed on the transition network, to speed up computation.
- For each of the eighteen clouds, each of the nine pattern-violating tokens not included in the cloud was successively selected as input, for which PEBLS generated an output.
  - Also for each of the 10 pattern-conforming tokens, using a leave-one-out procedure in constructing the clouds.
- We measured mean energy during the medial consonant of the outputs. High values reflect a spirantized output, whereas low values reflect stop closure.
The audio signals were preprocessed into frames of 13 mel-frequency cepstral coefficients (MFCCs).

The similarity drop-off parameter $c$ was set to 30.

In addition, a similarity threshold of 0.1 was imposed on the transition network, to speed up computation.

For each of the eighteen clouds, each of the nine pattern-violating tokens not included in the cloud was successively selected as input, for which PEBLS generated an output.

Also for each of the 10 pattern-conforming tokens, using a leave-one-out procedure in constructing the clouds.

We measured mean energy during the medial consonant of the outputs. High values reflect a spirantized output, whereas low values reflect stop closure.
Design details

- The audio signals were preprocessed into frames of 13 mel-frequency cepstral coefficients (MFCCs).
- The similarity drop-off parameter $c$ was set to 30.
- In addition, a similarity threshold of 0.1 was imposed on the transition network, to speed up computation.
- For each of the eighteen clouds, each of the nine pattern-violating tokens not included in the cloud was successively selected as input, for which PEBLS generated an output.
  - Also for each of the 10 pattern-conforming tokens, using a leave-one-out procedure in constructing the clouds.
- We measured mean energy during the medial consonant of the outputs. High values reflect a spirantized output, whereas low values reflect stop closure.
Design details

- The audio signals were preprocessed into frames of 13 mel-frequency cepstral coefficients (MFCCs).
- The similarity drop-off parameter \( c \) was set to 30.
- In addition, a similarity threshold of 0.1 was imposed on the transition network, to speed up computation.
- For each of the eighteen clouds, each of the nine pattern-violating tokens not included in the cloud was successively selected as input, for which PEBLS generated an output.
  - Also for each of the 10 pattern-conforming tokens, using a leave-one-out procedure in constructing the clouds.
- We measured mean energy during the medial consonant of the outputs. High values reflect a spirantized output, whereas low values reflect stop closure.
Design details

- The audio signals were preprocessed into frames of 13 mel-frequency cepstral coefficients (MFCCs).
- The similarity drop-off parameter $c$ was set to 30.
- In addition, a similarity threshold of 0.1 was imposed on the transition network, to speed up computation.
- For each of the eighteen clouds, each of the nine pattern-violating tokens not included in the cloud was successively selected as input, for which PEBLS generated an output.
  - Also for each of the 10 pattern-conforming tokens, using a leave-one-out procedure in constructing the clouds.
- We measured mean energy during the medial consonant of the outputs. High values reflect a spirantized output, whereas low values reflect stop closure.
The audio signals were preprocessed into frames of 13 mel-frequency cepstral coefficients (MFCCs).

The similarity drop-off parameter $c$ was set to 30.

In addition, a similarity threshold of 0.1 was imposed on the transition network, to speed up computation.

For each of the eighteen clouds, each of the nine pattern-violating tokens not included in the cloud was successively selected as input, for which PEBLS generated an output.

- Also for each of the 10 pattern-conforming tokens, using a leave-one-out procedure in constructing the clouds.

We measured mean energy during the medial consonant of the outputs. High values reflect a spirantized output, whereas low values reflect stop closure.
Outline

1. Background
2. PEBLS
   - Framing the problem
   - DTW
   - PEBLS: intra-cloud transition matrix
   - Confidence-sensitive alignment
3. Experiment I: simple output generation
   - Questions and method
   - Results and discussion
4. Experiment II: Iterative production
5. Conclusions
A few illustrative spectrograms

a. input: \[ækæ\]

b. resulting PEBLS output

c. input: [ext]
d. resulting PEBLS output
General results
Mean energy of medial consonants in PEBLS outputs

![Graph showing mean energy of medial consonants in PEBLS outputs](image-url)
Discussion

- Broadly speaking, the results show generalization of the allophonic spirantization pattern instantiated in each cloud:
  - In some words (/æ_æ/, /i_i/, /æ_s/, /e_t/, /s_æ/), the outputs uniformly adhere to the pattern,
  - whereas in others, the outputs vary in their pattern conformity.
- In the less interesting case of selection of pattern-conforming inputs, the outputs (not shown here) all uniformly conform to the pattern.
Broadly speaking, the results show generalization of the allophonic spirantization pattern instantiated in each cloud:

- In some words (/æ_æ/, /i_i/, /æ_s/, /e_t/, /s_æ/), the outputs uniformly adhere to the pattern,
- whereas in others, the outputs vary in their pattern conformity.

- In the less interesting case of selection of pattern-conforming inputs, the outputs (not shown here) all uniformly conform to the pattern.
Discussion

- Broadly speaking, the results show generalization of the allophonic spirantization pattern instantiated in each cloud:
  - In some words (\(\text{/æ_æ/}, \text{/i_i/}, \text{/æ_s/}, \text{/e_t/}, \text{/s_æ/}\)), the outputs uniformly adhere to the pattern,
  - whereas in others, the outputs vary in their pattern conformity.
- In the less interesting case of selection of pattern-conforming inputs, the outputs (not shown here) all uniformly conform to the pattern.
Discussion

- Broadly speaking, the results show generalization of the allophonic spirantization pattern instantiated in each cloud:
  - In some words (/æ_æ/, /i_i/, /æ_s/, /e_t/, /s_æ/), the outputs uniformly adhere to the pattern,
  - whereas in others, the outputs vary in their pattern conformity.

- In the less interesting case of selection of pattern-conforming inputs, the outputs (not shown here) all uniformly conform to the pattern.
Hypothesis

- As the system generates outputs iteratively, the word type should show a progression toward uniform adherence to the pattern,
- i.e. pattern entrenchment.
Hypothesis

- As the system generates outputs iteratively, the word type should show a progression toward uniform adherence to the pattern,
- i.e. pattern entrenchment.
Iteration with PEBLS’ current input selection method, however, is problematic:

- introduces new tokens in the cloud with particular frames, or even long sequences of frames, which may *exactly* match frames of the input.
- In PEBLS, exact matches seem to trump confidence sensitivity.
- Circumvented by adding a modicum of random noise to each output as it is appended to the cloud.

We tested PEBLS’ iterative productions for /e_e/ (one of the still variable clouds in Experiment 1).
Method

- Iteration with PEBLS’ current input selection method, however, is problematic:
  - introduces new tokens in the cloud with particular frames, or even long sequences of frames, which may *exactly* match frames of the input.
  - In PEBLS, exact matches seem to trump confidence sensitivity.
  - Circumvented by adding a modicum of random noise to each output as it is appended to the cloud.
- We tested PEBLS’ iterative productions for /e_e/ (one of the still variable clouds in Experiment 1).
Method

- Iteration with PEBLS’ current input selection method, however, is problematic:
  - introduces new tokens in the cloud with particular frames, or even long sequences of frames, which may exactly match frames of the input.
  - In PEBLS, exact matches seem to trump confidence sensitivity.
  - Circumvented by adding a modicum of random noise to each output as it is appended to the cloud.

- We tested PEBLS’ iterative productions for /e_e/ (one of the still variable clouds in Experiment 1).
Method

- Iteration with PEBLS’ current input selection method, however, is problematic:
  - introduces new tokens in the cloud with particular frames, or even long sequences of frames, which may exactly match frames of the input.
  - In PEBLS, exact matches seem to trump confidence sensitivity.
  - Circumvented by adding a modicum of random noise to each output as it is appended to the cloud.

- We tested PEBLS’ iterative productions for /e_e/ (one of the still variable clouds in Experiment 1).
Method

- Iteration with PEBLS’ current input selection method, however, is problematic:
  - introduces new tokens in the cloud with particular frames, or even long sequences of frames, which may *exactly* match frames of the input.
  - In PEBLS, exact matches seem to trump confidence sensitivity.
  - Circumvented by adding a modicum of random noise to each output as it is appended to the cloud.

- We tested PEBLS’ iterative productions for /e_e/ (one of the still variable clouds in Experiment 1).
Results
Mean energy of medial consonant in iterative productions of /eke/
Discussion

- The results show intermittent stop outputs which begin to taper off after about 100 iterations,
  - ceasing altogether after the 411th iteration,
  - and continuing with only fricative outputs for 200 iterations thereafter.
- We infer that, for this word, after these iterations, the spirantization allophone has become obligatory.
The results show intermittent stop outputs which begin to taper off after about 100 iterations,

- ceasing altogether after the 411\textsuperscript{th} iteration,
- and continuing with only fricative outputs for 200 iterations thereafter.

We infer that, for this word, after these iterations, the spirantization allophone has become obligatory.
The results show intermittent stop outputs which begin to taper off after about 100 iterations, 
- ceasing altogether after the 411th iteration,
- and continuing with only fricative outputs for 200 iterations thereafter.

We infer that, for this word, after these iterations, the spirantization allophone has become obligatory.
The results show intermittent stop outputs which begin to taper off after about 100 iterations,

- ceasing altogether after the 411th iteration,
- and continuing with only fricative outputs for 200 iterations thereafter.

We infer that, for this word, after these iterations, the spirantization allophone has become obligatory.
Conclusions

- The notion of pattern entrenchment in exemplar dynamics has been a central claim of Exemplar Theory. It is the sum and substance of the ET story on where phonology comes from – how categorical, stable (i.e. quasi-symbolic) behaviour arises from numerical signals.
- PEBLS provides the first explicit model of this emergent effect with real speech signals.
- The next step in this research programme is to show generalization outside the word class.
Conclusions

- The notion of pattern entrenchment in exemplar dynamics has been a central claim of Exemplar Theory. It is the sum and substance of the ET story on where phonology comes from – how categorical, stable (i.e. quasi-symbolic) behaviour arises from numerical signals.

- PEBLS provides the first explicit model of this emergent effect with real speech signals.

- The next step in this research programme is to show generalization outside the word class.
The notion of pattern entrenchment in exemplar dynamics has been a central claim of Exemplar Theory. It is the sum and substance of the ET story on where phonology comes from – how categorical, stable (i.e. quasi-symbolic) behaviour arises from numerical signals.

PEBLS provides the first explicit model of this emergent effect with real speech signals.

The next step in this research programme is to show generalization outside the word class.
Parallels to OT

- Inasmuch as PEBLS computes a global optimization for the output, there exist deep parallels to Optimality Theory.
  - PEBLS' alignment of input to cloud is analogous to OT enforcement of correspondence constraints.
  - A more elaborated version of PEBLS would include soft constraints reflecting phonetic pressures as part of the optimization criterion, analogous to OT markedness constraints.
  - In PEBLS then, as in OT, phonological patterns would arise from conflict between constraints favouring current patterns (including patterns within the word-class, as with IO-faithfulness), and constraints favouring phonetic naturalness.
- PEBLS, however, computes over numeric signals rather than symbolic representations, thus providing a seamless phonetics-phonology interface.

Robert Kirchner, Roger K. Moore
Computing over exemplars
Parallels to OT

Inasmuch as PEBLS computes a global optimization for the output, there exist deep parallels to Optimality Theory.

- PEBLS’ alignment of input to cloud is analogous to OT enforcement of correspondence constraints.
- A more elaborated version of PEBLS would include soft constraints reflecting phonetic pressures as part of the optimization criterion, analogous to OT markedness constraints.
- In PEBLS then, as in OT, phonological patterns would arise from conflict between constraints favouring current patterns (including patterns within the word-class, as with IO-faithfulness), and constraints favouring phonetic naturalness.

- PEBLS, however, computes over numeric signals rather than symbolic representations, thus providing a seamless phonetics-phonology interface.

Robert Kirchner, Roger K. Moore

Computing over exemplars
Parallels to OT

Inasmuch as PEBLS computes a global optimization for the output, there exist deep parallels to Optimality Theory.

- PEBLS’ alignment of input to cloud is analogous to OT enforcement of correspondence constraints.
- A more elaborated version of PEBLS would include soft constraints reflecting phonetic pressures as part of the optimization criterion, analogous to OT markedness constraints.
- In PEBLS then, as in OT, phonological patterns would arise from conflict between constraints favouring current patterns (including patterns within the word-class, as with IO-faithfulness), and constraints favouring phonetic naturalness.

PEBLS, however, computes over numeric signals rather than symbolic representations, thus providing a seamless phonetics-phonology interface.

Robert Kirchner, Roger K. Moore
Computing over exemplars
Parallels to OT

- Inasmuch as PEBLS computes a global optimization for the output, there exist deep parallels to Optimality Theory.
  - PEBLS’ alignment of input to cloud is analogous to OT enforcement of correspondence constraints.
  - A more elaborated version of PEBLS would include soft constraints reflecting phonetic pressures as part of the optimization criterion, analogous to OT markedness constraints.
  - In PEBLS then, as in OT, phonological patterns would arise from conflict between constraints favouring current patterns (including patterns within the word-class, as with IO-faithfulness), and constraints favouring phonetic naturalness.

- PEBLS, however, computes over numeric signals rather than symbolic representations, thus providing a seamless phonetics-phonology interface.

Robert Kirchner, Roger K. Moore  Computing over exemplars
Parallels to OT

- Inasmuch as PEBLS computes a global optimization for the output, there exist deep parallels to Optimality Theory.
  - PEBLS’ alignment of input to cloud is analogous to OT enforcement of correspondence constraints.
  - A more elaborated version of PEBLS would include soft constraints reflecting phonetic pressures as part of the optimization criterion, analogous to OT markedness constraints.
  - In PEBLS then, as in OT, phonological patterns would arise from conflict between constraints favouring current patterns (including patterns within the word-class, as with IO-faithfulness), and constraints favouring phonetic naturalness.
- PEBLS, however, computes over numeric signals rather than symbolic representations, thus providing a seamless phonetics-phonology interface.
Available at http://www.ualberta.ca/~kirchner/PEBLS


