Learning acoustic features for English stops with graph-based dimensionality reduction

Patrick F. Reidy, Mary E. Beckman, Jan Edwards, Benjamin Munson, Allison A. Johnson

Overview
The computation of spectral features that cue segmental contrasts is a process of dimensionality reduction. Traditional approaches accomplish this reduction by mapping a high-dimensional observation (e.g., a spectrum) to a small number of pre-determined features (e.g., spectral moments; Forrest et al., 1988). Such approaches fail to exploit the distributional structure of the observations in the high-dimensional space and typically ignore superposing relationships among the observations, such as the word in which the segment occurs.

This study adapts the Laplacian Eigenmaps algorithm (Belkin & Niyogi, 2003; Bengio et al., 2003) to learn acoustic features for /t/ versus /k/, consonants that contrast in terms of spectral shape and that differentially exhibit vowel-contextual variation in their spectral shape (see Fig. 1). The algorithm constructs a graph that simultaneously represents the high-dimensional structure of excitation patterns computed from a talker’s productions, and aligns lexical correspondences between talkers. A function that embeds the excitation patterns into a two-dimensional feature space is learned by computing the eigenvectors of the constructed graph.

Speech Production Data

21 adults (10 women, 11 men) completed a picture-prompted word repetition task.
Two lists of words were used to elicit a variety of target consonants. Each list contained 32 words in which a target /t/ or /k/ occurred word-initially before a vowel (see footer at bottom for the start-initial words in the two lists).
Participants A50–A65 completed Lists A and B; participants A66–A70, only List A.
Training set: List A productions by participants A50–A65 (N = 403).
Test sets: List B from A50–A65 (N = 447); List A from A66–A70 (N = 156).
Multitaper spectra were estimated from 25-ms windows around stop bursts, and then passed through an auditory (gammatone) filter bank, yielding excitation patterns.

Laplacian Eigenmaps Algorithm
1. Let \( X = \{x_1, \ldots, x_n\} \) be the training set of 493 excitation patterns (361-dimensional vectors).
Each \( x_i \) is pre-processed to sum to 1, so that it may be treated as a probability mass function.
2. Define a similarity function \( \mathcal{S}_{ij} \) on \( X \) in terms of Kulback-Leibler divergence \( D_{KL} \) (see Fig. 2).
3. Define a function \( \mathcal{W} \) on \( X \) that induces a graph. Nodes correspond to observations in \( X \) (see diagram below), where \( (x_i, x_j) \) versus \( (x_k, x_l) \) represent productions by different talkers. Edges connect nodes corresponding either to productions of the same talker (solid lines) or to productions of the same target word by different talkers (dashed lines). Edges weight encode similarity between excitation patterns. Parameter \( \mu \in (0, 1) \) adjusts the balance between preserving the structure of each talker’s production-space and aligning multiple talkers’ production-spaces.

\[
\mathcal{W}(x_i, x_j) = \begin{cases} 
(1 - \mu) \mathcal{S}(x_i, x_j) & \text{if \ (word(x_i) = word(x_j)) \ and \ (talker(x_i) \neq talker(x_j))} \\
\mu \mathcal{S}(x_i, x_j) & \text{if \ (talker(x_i) = talker(x_j))} \\
0 & \text{otherwise}.
\end{cases}
\]

4. Construct the graph’s weighted adjacency matrix \( A \), degree matrix \( D \), and Laplacian matrix \( L \).

\[
A = \mathcal{W}(x_i, x_j) \quad D_i = \sum_j A_{ij} \quad L = D - A.
\]

5. Solve the generalized eigenvalue problem \( \lambda \mathcal{L} \mathcal{X} = \mu \mathcal{X} \).
The eigenvectors \( \mathcal{X} \) correspond to the two least, non-zero eigenvalues embed \( X \) into 2-dimensional space:

\[
x_i \rightarrow (\lambda_1[x_i], \lambda_2[x_i])
\]

6. Extend eigenvectors \( \lambda_1, \lambda_2 \) to eigenfunctions \( \lambda_1, \lambda_2 \). The projection \( \lambda_1[x_i], \lambda_2[x_i] \) of a data point \( x \) onto dimension \( k \) of the embedding is a linear combination of the components of \( \lambda_k \).

\[
\lambda_k[x_i] = \sum_j \lambda_k[x_j] \cdot \mathcal{W}(x_j, x_i)
\]

Discussion and Future Directions

The two-dimensional embedding that is learned by Laplacian Eigenmaps reflects well-established articulatory constriction features. \( \lambda_1 \) distinguishes /t/ versus back-vs.-front;/k/, reflecting place of constriction (anterior versus posterior); \( \lambda_2 \) distinguishes /t/ versus front-/k/, reflecting tongue shape (apical versus damped).

We plan to extend this method to develop dynamic spectral features that model the transition from a stop burst to a vowel (see Nosari & Zarahn, 1991).

References